Statistical outputs

```r
#----------------------------
#import libraries
library(ggspatial)
library(cowplot)
library(stringr)
library(dplyr)
library(raster)
library(sf)
library(maptools)
library(stars)
library(lubridate)
library(data.table)
library(ggplot2)
library(purrr)
library(lme4)
library(lmerTest)
library(effects)
library(introdataviz) # split violin plots
library(extrafont)
```

Prepare dataset

```r
#--------------------------
options(scipen=999)
Sys.setenv(TZ='UTC') # change time zone for session
setwd("D:/OneDrive - NIOZ/")
#setwd("C:/Users/cbeardsworth/OneDrive - NIOZ")

ele <- raster("2_Reference Data/Elevation/griend_bathy_2015-2020.tif")
SA <- read.csv("2_Reference Data/Water/spacebywaterlevel.csv")
griend_poly <- read_sf("2_Reference Data/Griend/Griend_2015-2020Bathy_100NAP.gpkg")
meta <- read.csv("2_Reference Data/Bird Metadata/2018-2022_islandica.csv") %>%
            dplyr::select(release_ts, tag, captivity, age)

waterdn <- read.csv("2_Reference Data/Water/2017-2022_waterlevel_under50_propdaynight.csv") %>% # propo
    mutate(total_light = prop_day + prop_civil,
           total_dark = prop_naut + prop_astro + prop_night,
           tod_summary = case_when(total_light >= 0.70 ~ "day", #tide majority in day
                                   total_light <0.70 & total_dark <0.70 ~ "twilight",
                                   total_dark >=0.70 ~ "night")) #tide majority at night

tides <- read.csv("2_Reference Data/Water/Griend/Wide/HLH/allYears-tidalPattern-HLH-griend_Wide-UTC.csv
    mutate(year = substring(tideID,1,4),
```

```r
           yearly_id = as.numeric(substring(tideID, 6,8)),
           tideID = as.numeric(sub("_", "", tideID)),
           high_start_time = as.POSIXct(high_start_time),
           date = as.Date(high_start_time, tz="UTC"),
           month = month(high_start_time)) %>%
    left_join(SA, by = c("low_level"="waterlevel")) %>% #space available in the tide
    left_join(waterdn, by="tideID") %>%
    filter(month>=9, tod_summary!="twilight" & start_tod == tod_summary) %>% #choose only after Septemb
    rename(tide_tod = start_tod)

#choose residence patches that are not roosting (<50NAP), adults only, not our birds that were in capti
df <- fread("1_WATLAS/Data/res_patches/2018-2022_patches.csv") %>%
    as.data.frame() %>%
    filter(!tag  %in% c(911,923), captivity == "no", age==3) %>%   #choose birds for analysis
    mutate(tod_start = ifelse(tod_start %in% c("day", "civil twilight"), "day", "night"),
           datetime_start_UTC = as.POSIXct(datetime_start_UTC, tz = "UTC"),
           month_num = str_pad(lubridate::month(datetime_start_UTC),2,pad="0"),
           month_name = lubridate::month(datetime_start_UTC, label=T),
           month = paste(month_num, month_name, sep="_"),
           tag= str_pad(tag, 4, pad = "0"),
           bird_elevation_start = raster::extract(ele, cbind(x_start, y_start)), # get elevation for wh
           roost_start = ifelse(bird_elevation_start>=50, 1, 0),
           bird_elevation_end = raster::extract(ele, cbind(x_end, y_end)),
           roost_end = ifelse(bird_elevation_end>=50, 1, 0),
           roost_site = ifelse(roost_start ==1 | roost_end == 1, 1,0)) %>%
    filter(roost_site == 0,
           tideID %in% tides$tideID) %>%  # only choose tides that started at the same time of day as t
    left_join(., tides[,c("tideID", "tide_tod")], by = c("tideID"= "tideID")) %>%
    st_as_sf(coords = c("x_start","y_start"), crs = 32631, remove = F) %>%
    mutate(dist_from_griendpoly = as.numeric(st_distance(., griend_poly)))

df$geometry <- NULL

# Calculate duration tracked per ind, per tide.

bird_tides <- df %>%
    group_by(year, tideID, tag, tide_tod) %>%
    summarise(sum_duration_min = sum(duration_s)/60, ##low tide durations, we calculate the sum of the 
              furthest_patch = max(dist_from_griendpoly)) %>% ##calculate furthest distance from griend
    filter(sum_duration_min >=240) %>%  # which birds on which tides should be analysed.
    left_join(tides[,c("tideID","low_level")], by = "tideID") #add low level for analysis on distance f
```

```
## `summarise()` has grouped output by 'year', 'tideID', 'tag'. You can override
## using the `.groups` argument.
```

```r
min_tides <- 5 ## choose min tides
tides_table <- as.data.frame.matrix(table(bird_tides$tag,bird_tides$tide_tod)) %>% ##which birds have a
    filter(day>= min_tides, night >= min_tides)

bird_tides <- bird_tides %>%
    filter(tag %in% row.names(tides_table)) %>%  ##choose only birds that had minimum tides tracked for
    mutate(year = as.factor(year),
           tag = as.factor(tag))
```

```r
# sample sizes

table(bird_tides$year,bird_tides$tide_tod) ##total tides tracked in day/night
```

```
##
##        day night
##   2018 386   262
##   2019 365   195
##   2020  56    74
##   2021 416   189
##   2022 239   124
```

```r
tide_sample_size <- unique(as.data.frame(bird_tides[, c("tag", "tideID", "tide_tod")])) ##number of tid
tide_sample_size <- as.data.frame.matrix(table(tide_sample_size$tag, tide_sample_size$tide_tod))
tide_sample_size$dif <- tide_sample_size$day - tide_sample_size$night
summary(tide_sample_size$day) ## how many day tides were birds tracked for
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00    9.75   13.00   13.54   17.00   27.00
```

```r
summary(tide_sample_size$night) ## how many night tides were they tracked for
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.000   6.000   7.000   7.815  10.000  16.000
```

```r
summary(tide_sample_size$dif) ##what were the differences within individuals for tracking day or night
```

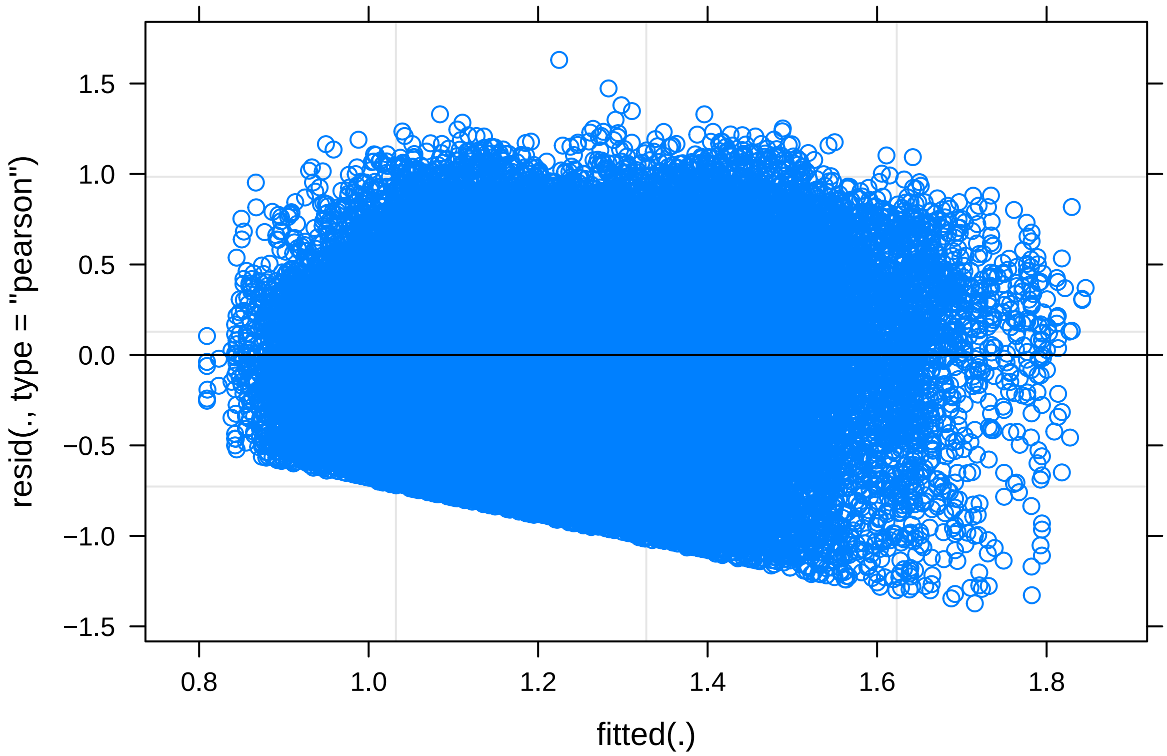
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -7.000   2.750   5.000   5.722   9.000  17.000
```

```r
sample_size <- unique(as.data.frame(bird_tides[, c("year", "tag")]))
table(sample_size$year)
```

```
##
## 2018 2019 2020 2021 2022
##   36   23    7   24   18
```

Question 1: Do knots stay longer in residence patches at night?

```r
model_dur <- lmer(data=df, log_duration_min ~ tod_start*inout + (1|year) + (1|tag) + (1|tideID))
plot(model_dur)
```



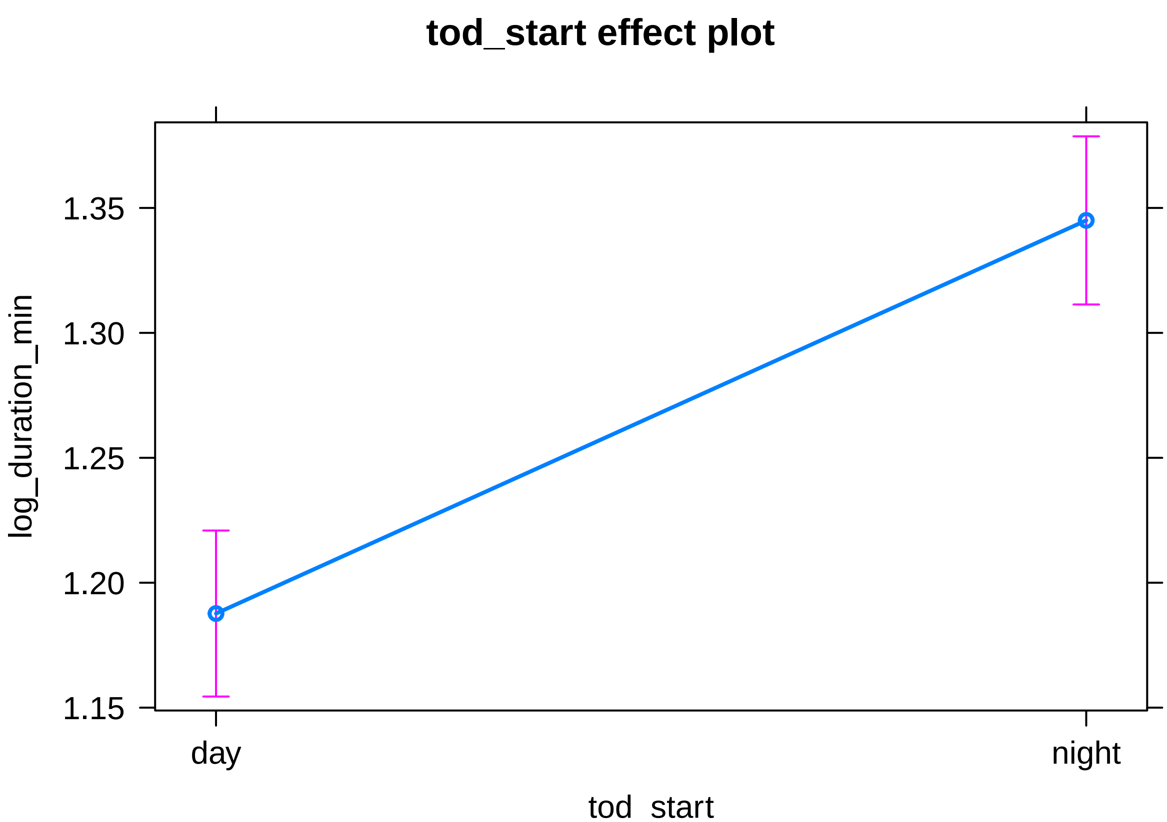

```r
#anova(model_dur, type="III")
summary(model_dur)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: log_duration_min ~ tod_start * inout + (1 | year) + (1 | tag) +
##     (1 | tideID)
##    Data: df
##
## REML criterion at convergence: 75396.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1085 -0.7212  0.0305  0.7293  3.6912
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  tag      (Intercept) 0.001715 0.04141
##  tideID   (Intercept) 0.011637 0.10788
##  year     (Intercept) 0.001108 0.03329
##  Residual             0.195066 0.44166
## Number of obs: 61879, groups:  tag, 378; tideID, 273; year, 5
##
## Fixed effects:
##                                  Estimate   Std. Error           df t value
```

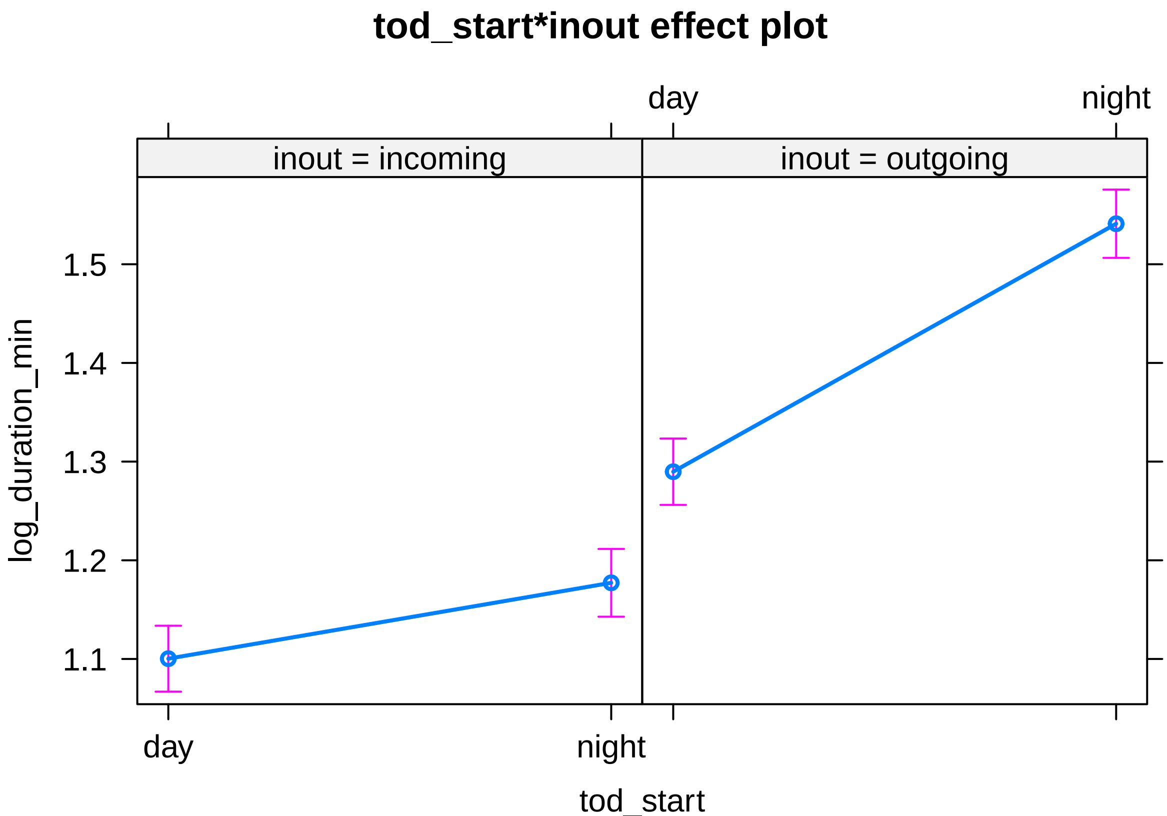
```
## (Intercept)                         1.100294     0.017042     4.246463   64.56
## tod_startnight                      0.076882     0.006671 49242.769206   11.53
## inoutoutgoing                       0.189473     0.004527 59652.472833   41.86
## tod_startnight:inoutoutgoing        0.174416     0.009285 51743.020369   18.78
##                                                Pr(>|t|)
## (Intercept)                               0.000000162 ***
## tod_startnight               < 0.0000000000000002 ***
## inoutoutgoing                < 0.0000000000000002 ***
## tod_startnight:inoutoutgoing < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) td_str inttgn
## td_strtnght -0.123
## inoutoutgng -0.106  0.282
## td_strtngh:  0.050 -0.564 -0.580
```

```r
plot(effect("tod_start", model_dur))
```

```
## NOTE: tod_start is not a high-order term in the model
```

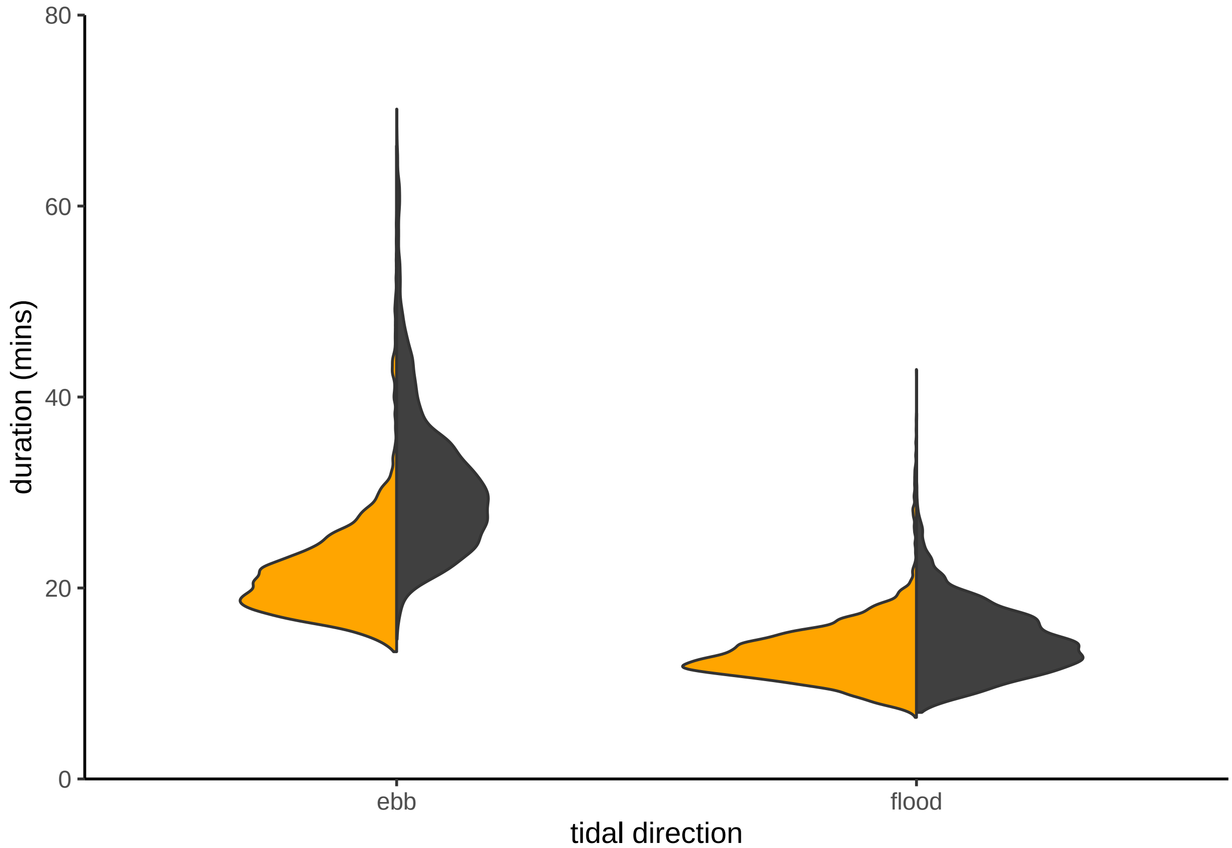



```r
plot(effect("tod_start*inout", model_dur))
```




```
duration_df <- df[,c("tod_start", "tag", "year", "tideID","inout")]
duration_df$log_duration_min <- predict(model_dur, newdata=duration_df,type="response")
duration_df$duration_min <- 10^duration_df$log_duration_min
duration_df[duration_df$inout == "incoming",]$inout <- "flood"
duration_df[duration_df$inout == "outgoing",]$inout <- "ebb"

p1 <- ggplot(data= duration_df)+
  geom_split_violin(aes(y = duration_min, x = inout, fill=tod_start))+
  scale_fill_manual(values=c("orange","grey25"))+
  scale_y_continuous(expand=c(0,0), limits = c(0,80), name = "duration (mins)")+
  scale_x_discrete(name = "tidal direction")+
  theme_classic()+
  theme(legend.position="none")
p1
```



Question 2: Do red knots stay closer to Griend at night

```
bird_tides$low_level_scaled <- scale(bird_tides$low_level, center = T, scale = F) # center subtracts th
scale_c <- attributes(bird_tides$low_level_scaled)$`scaled:center`

model_furth <- lmer(data=bird_tides, furthest_patch ~ tide_tod + low_level_scaled + (1|year) + (1|tag))
anova(model_furth, type="III")
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                     Sum Sq   Mean Sq NumDF  DenDF  F value              Pr(>F)
## tide_tod           5351605   5351605     1 2209.7   4.9797             0.02575
## low_level_scaled 108128481 108128481     1 2220.6 100.6139 < 0.0000000000000002
##
## tide_tod         *
## low_level_scaled ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model_furth)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
```

```
## Formula: furthest_patch ~ tide_tod + low_level_scaled + (1 | year) + (1 |
##     tag)
##    Data: bird_tides
##
## REML criterion at convergence: 38771.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7728 -0.7304 -0.0127  0.7121  3.4371
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  tag      (Intercept)  359423   599.5
##  year     (Intercept)   35376   188.1
##  Residual             1074687  1036.7
## Number of obs: 2306, groups:  tag, 108; year, 5
##
## Fixed effects:
##                   Estimate Std. Error        df t value             Pr(>|t|)
## (Intercept)      2476.5585   108.7888    1.9585  22.765              0.00213
## tide_todnight    -101.6327    45.5441 2209.6783  -2.232              0.02575
## low_level_scaled   -7.0485     0.7027 2220.5898 -10.031 < 0.0000000000000002
##
## (Intercept)      **
## tide_todnight    *
## low_level_scaled ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) td_tdn
## tide_tdnght -0.161
## lw_lvl_scld -0.008  0.018
```

```r
coef(model_furth)
```

```
## $tag
##      (Intercept) tide_todnight low_level_scaled
## 0468    2508.978     -101.6327        -7.048547
## 0473    2659.612     -101.6327        -7.048547
## 0476    2527.111     -101.6327        -7.048547
## 0478    2762.021     -101.6327        -7.048547
## 0480    3264.720     -101.6327        -7.048547
## 0481    2808.178     -101.6327        -7.048547
## 0482    2387.197     -101.6327        -7.048547
## 0489    2744.610     -101.6327        -7.048547
## 0490    3461.988     -101.6327        -7.048547
## 0491    2212.894     -101.6327        -7.048547
## 0492    2160.173     -101.6327        -7.048547
## 0496    2185.666     -101.6327        -7.048547
## 0499    2777.101     -101.6327        -7.048547
## 0439    2596.058     -101.6327        -7.048547
## 0504    1839.022     -101.6327        -7.048547
## 0505    2344.420     -101.6327        -7.048547
```

```
## 0509    1691.708     -101.6327        -7.048547
## 0513    2016.918     -101.6327        -7.048547
## 0520    3168.061     -101.6327        -7.048547
## 0526    2248.638     -101.6327        -7.048547
## 0534    2114.538     -101.6327        -7.048547
## 0541    3414.263     -101.6327        -7.048547
## 0542    1996.733     -101.6327        -7.048547
## 0543    3512.853     -101.6327        -7.048547
## 0551    2143.231     -101.6327        -7.048547
## 0554    2101.732     -101.6327        -7.048547
## 0560    2584.319     -101.6327        -7.048547
## 0564    1879.060     -101.6327        -7.048547
## 0565    2109.248     -101.6327        -7.048547
## 0566    2024.321     -101.6327        -7.048547
## 0567    2124.031     -101.6327        -7.048547
## 0574    1825.500     -101.6327        -7.048547
## 0576    1396.632     -101.6327        -7.048547
## 0579    2500.310     -101.6327        -7.048547
## 0586    3012.482     -101.6327        -7.048547
## 0506    2333.252     -101.6327        -7.048547
## 0789    2674.446     -101.6327        -7.048547
## 0797    3634.237     -101.6327        -7.048547
## 0799    1893.514     -101.6327        -7.048547
## 0806    2437.492     -101.6327        -7.048547
## 0832    3500.571     -101.6327        -7.048547
## 0833    3711.241     -101.6327        -7.048547
## 0793    2579.637     -101.6327        -7.048547
## 0800    2834.792     -101.6327        -7.048547
## 0804    2371.071     -101.6327        -7.048547
## 0807    3190.004     -101.6327        -7.048547
## 0815    1670.828     -101.6327        -7.048547
## 0828    2244.890     -101.6327        -7.048547
## 0835    1705.559     -101.6327        -7.048547
## 0843    2735.055     -101.6327        -7.048547
## 0844    1084.302     -101.6327        -7.048547
## 0849    2675.874     -101.6327        -7.048547
## 0850    1468.993     -101.6327        -7.048547
## 0854    3463.194     -101.6327        -7.048547
## 0859    1395.734     -101.6327        -7.048547
## 0814    2110.789     -101.6327        -7.048547
## 0816    2983.261     -101.6327        -7.048547
## 0860    3023.756     -101.6327        -7.048547
## 0916    1221.049     -101.6327        -7.048547
## 2037    2500.018     -101.6327        -7.048547
## 2117    2606.260     -101.6327        -7.048547
## 2133    2495.148     -101.6327        -7.048547
## 2141    2756.147     -101.6327        -7.048547
## 2167    3103.279     -101.6327        -7.048547
## 2020    3150.208     -101.6327        -7.048547
## 2110    3117.866     -101.6327        -7.048547
## 2342    2585.461     -101.6327        -7.048547
## 2348    3185.935     -101.6327        -7.048547
## 2384    2646.504     -101.6327        -7.048547
## 2385    2677.435     -101.6327        -7.048547
```

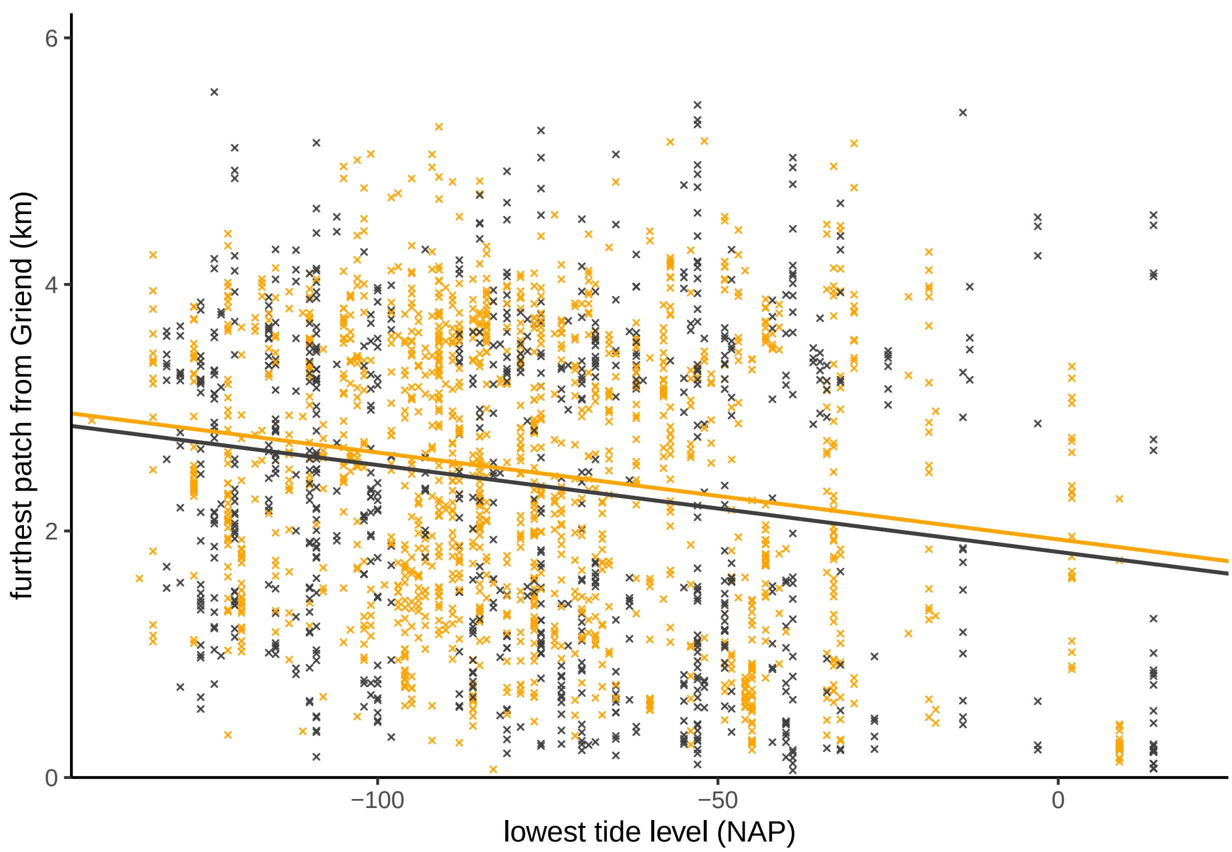
```
## 2389    2835.195     -101.6327        -7.048547
## 2390    2189.823     -101.6327        -7.048547
## 2392    2042.299     -101.6327        -7.048547
## 2395    2291.672     -101.6327        -7.048547
## 2396    2764.103     -101.6327        -7.048547
## 2398    2781.367     -101.6327        -7.048547
## 2410    2570.453     -101.6327        -7.048547
## 2440    2392.353     -101.6327        -7.048547
## 2449    2931.756     -101.6327        -7.048547
## 2452    1910.381     -101.6327        -7.048547
## 2454    2370.533     -101.6327        -7.048547
## 2455    2684.766     -101.6327        -7.048547
## 2462    2742.172     -101.6327        -7.048547
## 2465    2068.797     -101.6327        -7.048547
## 2482    2544.881     -101.6327        -7.048547
## 2490    1937.998     -101.6327        -7.048547
## 2493    2306.867     -101.6327        -7.048547
## 2494    2972.896     -101.6327        -7.048547
## 2502    1649.814     -101.6327        -7.048547
## 2476    2164.244     -101.6327        -7.048547
## 2701    2183.906     -101.6327        -7.048547
## 2714    2329.322     -101.6327        -7.048547
## 2715    2592.306     -101.6327        -7.048547
## 2716    3448.412     -101.6327        -7.048547
## 2718    3143.937     -101.6327        -7.048547
## 2721    2280.168     -101.6327        -7.048547
## 2722    2255.992     -101.6327        -7.048547
## 2724    3500.217     -101.6327        -7.048547
## 2709    2509.051     -101.6327        -7.048547
## 2742    2854.970     -101.6327        -7.048547
## 2748    2392.853     -101.6327        -7.048547
## 2758    1832.136     -101.6327        -7.048547
## 2788    1743.763     -101.6327        -7.048547
## 2800    2491.698     -101.6327        -7.048547
## 2815    1716.906     -101.6327        -7.048547
## 2833    2943.823     -101.6327        -7.048547
## 2840    2185.134     -101.6327        -7.048547
## 2835    2039.223     -101.6327        -7.048547
##
## $year
##      (Intercept) tide_todnight low_level_scaled
## 2018    2307.414     -101.6327        -7.048547
## 2019    2442.056     -101.6327        -7.048547
## 2020    2712.089     -101.6327        -7.048547
## 2021    2457.888     -101.6327        -7.048547
## 2022    2463.346     -101.6327        -7.048547
##
## attr(,"class")
## [1] "coef.mer"
```

```
intercept_day <- mean(c(coef(model_furth)$tag[,1],coef(model_furth)$year[,1])) - scale_c*coef(model_fur
intercept_night <- intercept_day + coef(model_furth)$tag[1,"tide_todnight"]
confint_day <- quantile(c(coef(model_furth)$tag[,1],coef(model_furth)$year[,1]), c(0.025, 0.975))
```

```r
slope <- coef(model_furth)$tag[1,"low_level_scaled"]

# Predict
furth_df <- unique(as.data.frame(bird_tides[,c("tide_tod", "tag", "year", "tideID","low_level_scaled")]
                    ))
furth_df$furthest_patch <- predict(model_furth, newdata=furth_df,type="response")
furth_df$low_level <- furth_df$low_level_scaled + mean(bird_tides$low_level)

p2 <- ggplot(data = bird_tides, aes(x=low_level, y = furthest_patch/1000, col = tide_tod))+
    geom_point(alpha = 0.9, pch = 4, size=0.7)+
    geom_abline(intercept = intercept_day/1000, slope = slope/1000, col = "orange", linewidth = 0.7)+
    geom_abline(intercept = intercept_night/1000, slope = slope/1000, col = "grey25", linewidth = 0.7)+
    scale_fill_manual(values=c("orange","grey25"))+
    scale_color_manual(values=c("orange","grey25"))+
    scale_y_continuous(expand=c(0,0), limits = c(0,6.2), name = "furthest patch from Griend (km)")+
    scale_x_continuous(expand=c(0,0), limits = c(-145,25), name = "lowest tide level (NAP)")+
    theme_classic()+
  theme(legend.position="none")
p2
```



Question 3: Do knots use smaller areas at night?

```
b_tides <- st_read("F:/OneDrive - Liverpool John Moores University/Synced files/Projects/1_NIOZ projects
  mutate(bird_tides = paste(tideID, id, sep="_"))
```

```
## Reading layer '2018-2022_bird_tides_multipolygons_min5tides' from data source
##   'F:\OneDrive - Liverpool John Moores University\Synced files\Projects\1_NIOZ projects\2_DayNight\da
##   using driver 'GPKG'
## Simple feature collection with 2306 features and 7 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 645999.5 ymin: 5900627 xmax: 656482.5 ymax: 5907160
## Projected CRS: WGS 84 / UTM zone 31N
```

```
area_ptide <- b_tides %>%
    arrange(id) %>%
    mutate(area_ptide = (area_km2/duration_tracked)*6*60) # area covered in a 6h low tide period.

area_ptide$geom <- NULL

model_area <- lmer(data=area_ptide, area_ptide ~ tide_tod + (1|year) + (1|id))
anova(model_area, type="III")
```

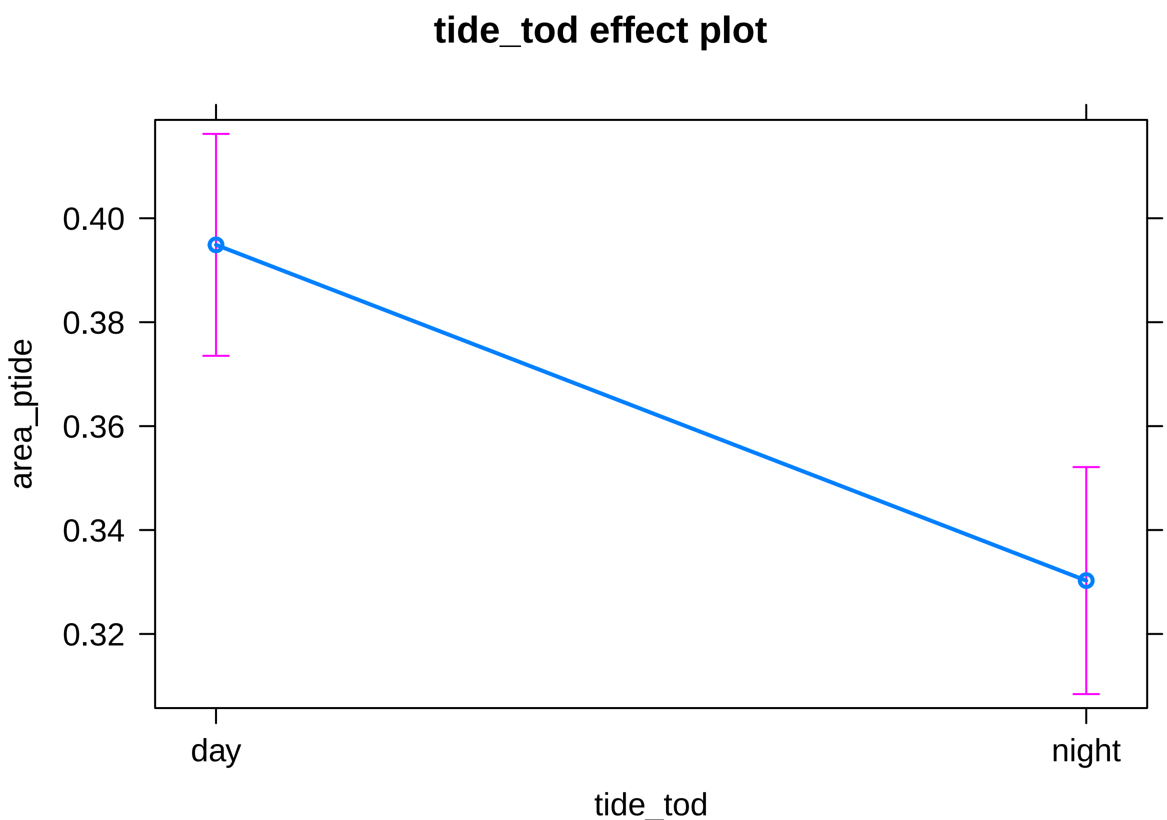
```
## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF  DenDF F value                 Pr(>F)
## tide_tod   2.17    2.17     1 2222.4  169.76 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model_area)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: area_ptide ~ tide_tod + (1 | year) + (1 | id)
##    Data: area_ptide
##
## REML criterion at convergence: -3347.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9664 -0.6220 -0.0242  0.6448  4.6356
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  id       (Intercept) 0.0016371 0.04046
##  year     (Intercept) 0.0004441 0.02107
##  Residual             0.0127828 0.11306
## Number of obs: 2306, groups:  id, 108; year, 5
##
## Fixed effects:
##                  Estimate  Std. Error          df t value             Pr(>|t|)
```

```
## (Intercept)       0.394877    0.010888    4.566029   36.27           0.000000852
## tide_todnight    -0.064600    0.004958 2222.361697  -13.03 < 0.0000000000000002
##
## (Intercept)    ***
## tide_todnight ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## tide_tdnght -0.177
```

```r
plot(effect("tide_tod", model_area))
```




```r
area_df <- area_ptide[,c("tide_tod", "id", "year")]
area_df$area_pred <- predict(model_area, newdata=area_df,type="response")

#mean predicted area for night
mean(area_df[area_df$tide_tod=="night",]$area_pred)
```

```
## [1] 0.3304045
```

```r
sd(area_df[area_df$tide_tod=="night",]$area_pred)
```

```
## [1] 0.03974139
```

```r
#mean predicted area during the day
mean(area_df[area_df$tide_tod=="day",]$area_pred)
```

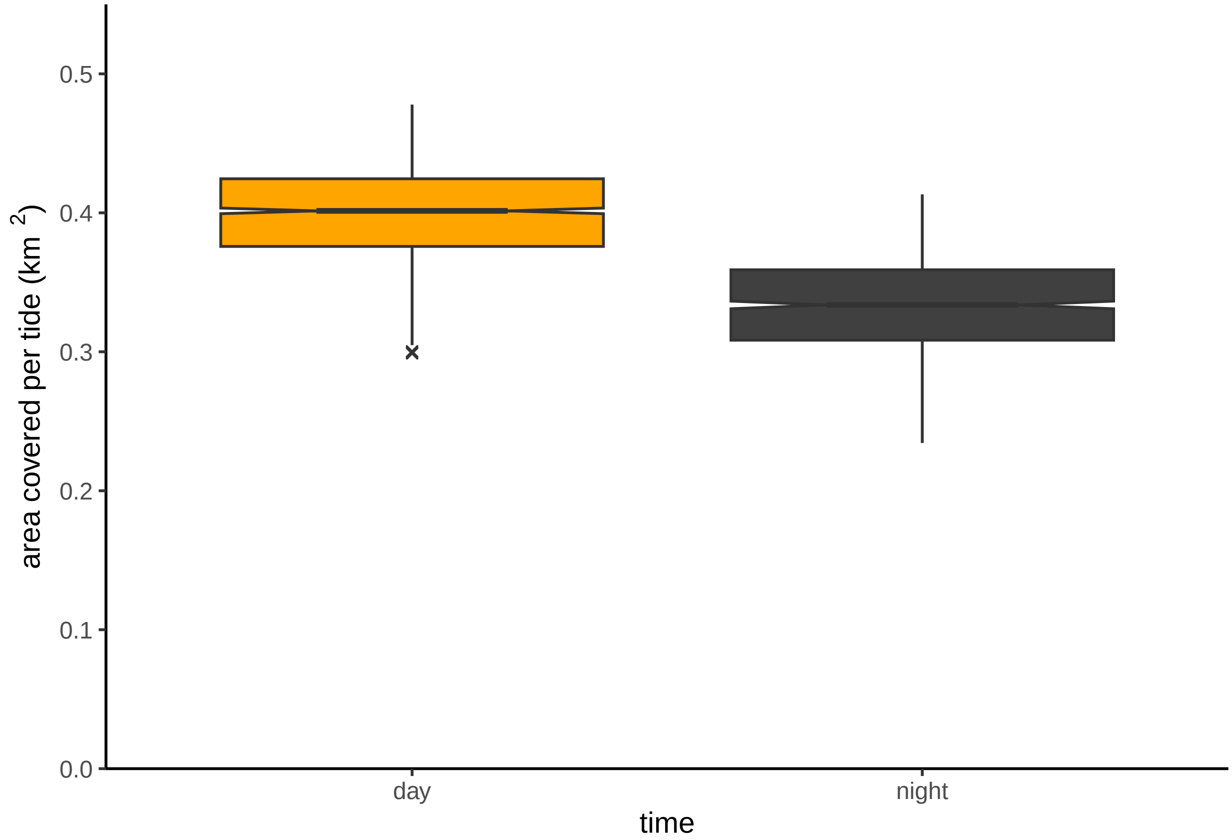
```
## [1] 0.3984496
```

```r
sd(area_df[area_df$tide_tod=="day",]$area_pred)
```

```
## [1] 0.03954141
```

```r
#predicted data
p3 <- ggplot(data = area_df, aes(x=tide_tod, y = area_pred, fill = tide_tod))+
    #geom_point(data = area_ptide, aes(x=tide_tod, y = area_ptide, fill = tide_tod), position = positio
    geom_boxplot(outlier.shape= 4, notch=T)+
    scale_fill_manual(values = c("orange","grey25"))+
    scale_x_discrete(name = "time")+
    scale_y_continuous(limits = c(0,0.55), expand = c(0,0), name = expression(paste("area covered per ti
    theme_classic()+
  theme(legend.position="none")

p3
```



Question 4: Do day and night areas overlap?

merge polygons for each bird (overlap) and calculate area used per tide

```
overlap <- st_read("F:/OneDrive - Liverpool John Moores University/Synced files/Projects/1_NIOZ project
    group_by(year, id, tide_tod) %>%
    summarise(geom= st_union(geom),
              duration = sum(duration_tracked),
              num_tides = n()) %>%
    ungroup() %>%
    arrange(id) %>%
    tidyr::pivot_wider(names_from = "tide_tod", values_from = c("duration", "num_tides", "geom")) %>%
    st_set_geometry("geom_day")  %>%
    group_by(id) %>%
    mutate(intersection = st_union(st_intersection(geom_day, geom_night)),
           area_day = as.numeric(st_area(geom_day))/10^6,
           area_night = as.numeric(st_area(geom_night))/10^6,
           area_intersect = as.numeric(st_area(intersection))/10^6,
           pc_overlap_day = as.numeric((area_intersect/area_day)*100),
           pc_overlap_night = as.numeric((area_intersect/area_night)*100),
           geom_day = NULL,
           geom_night = NULL,
           intersection = NULL) %>%
    as.data.frame()
```

```
## Reading layer '2018-2022_bird_tides_multipolygons_min5tides' from data source
##   'F:\OneDrive - Liverpool John Moores University\Synced files\Projects\1_NIOZ projects\2_DayNight\d
##   using driver 'GPKG'
## Simple feature collection with 2306 features and 7 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 645999.5 ymin: 5900627 xmax: 656482.5 ymax: 5907160
## Projected CRS: WGS 84 / UTM zone 31N

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

```
#mean total space use at night km2
mean(overlap$area_night)
```

```
## [1] 2.053526
```

```
sd(overlap$area_night)
```

```
## [1] 0.7581772
```

```
#mean number of tides at night per bird
mean(overlap$num_tides_night)
```

```
## [1] 7.814815
```

```r
sd(overlap$num_tides_night)
```

```
## [1] 2.510346
```

```r
#mean total space use in day km2
mean(overlap$area_day)
```

```
## [1] 4.483422
```

```r
sd(overlap$area_day)
```

```
## [1] 1.783366
```

```r
#mean number of tides in day per bird
mean(overlap$num_tides_day)
```

```
## [1] 13.53704
```

```r
sd(overlap$num_tides_day)
```

```
## [1] 5.116267
```

```r
#mean percentage overlap of day tide with night tide space
mean(overlap$pc_overlap_day)
```

```
## [1] 19.95742
```

```r
sd(overlap$pc_overlap_day)
```

```
## [1] 8.957541
```

```r
#mean percentage overlap of night tide with day tide space
mean(overlap$pc_overlap_night)
```

```
## [1] 40.95783
```

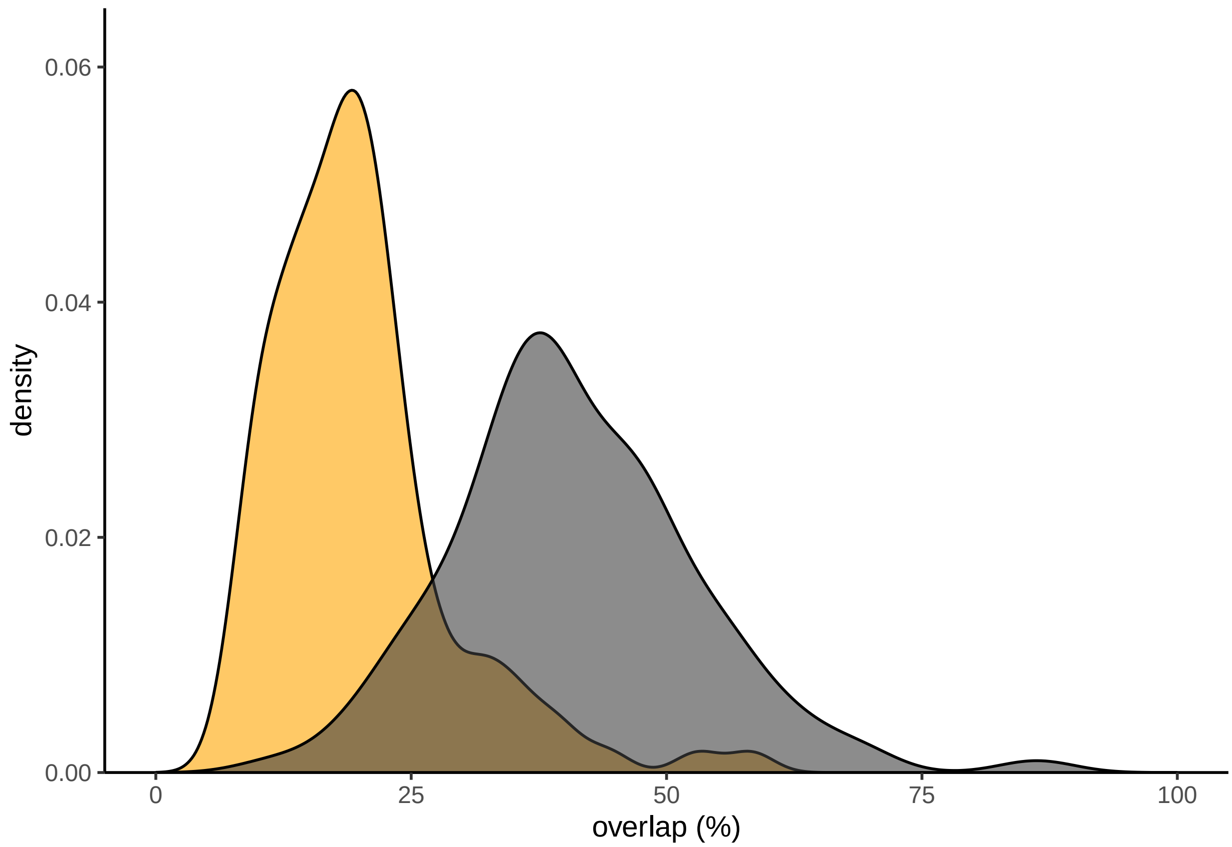
```r
sd(overlap$pc_overlap_night)
```

```
## [1] 11.90778
```

```r
#plot for individuals
overlap_long <- overlap %>%
    tidyr::pivot_longer(cols = c("pc_overlap_night","pc_overlap_day"),
                        names_to = "time",
                        values_to = "nonoverlap")
```

```r
p4 <- ggplot(overlap_long, aes(x = nonoverlap, fill = time))+
    geom_density(alpha = 0.6)+
    scale_fill_manual(values=c("orange","grey25"))+
    scale_x_continuous(limits = c(0,100), name = "overlap (%)")+
    scale_y_continuous(expand=c(0,0), limits = c(0,0.065))+
    theme_classic()+
  theme(legend.position="none")

p4
```



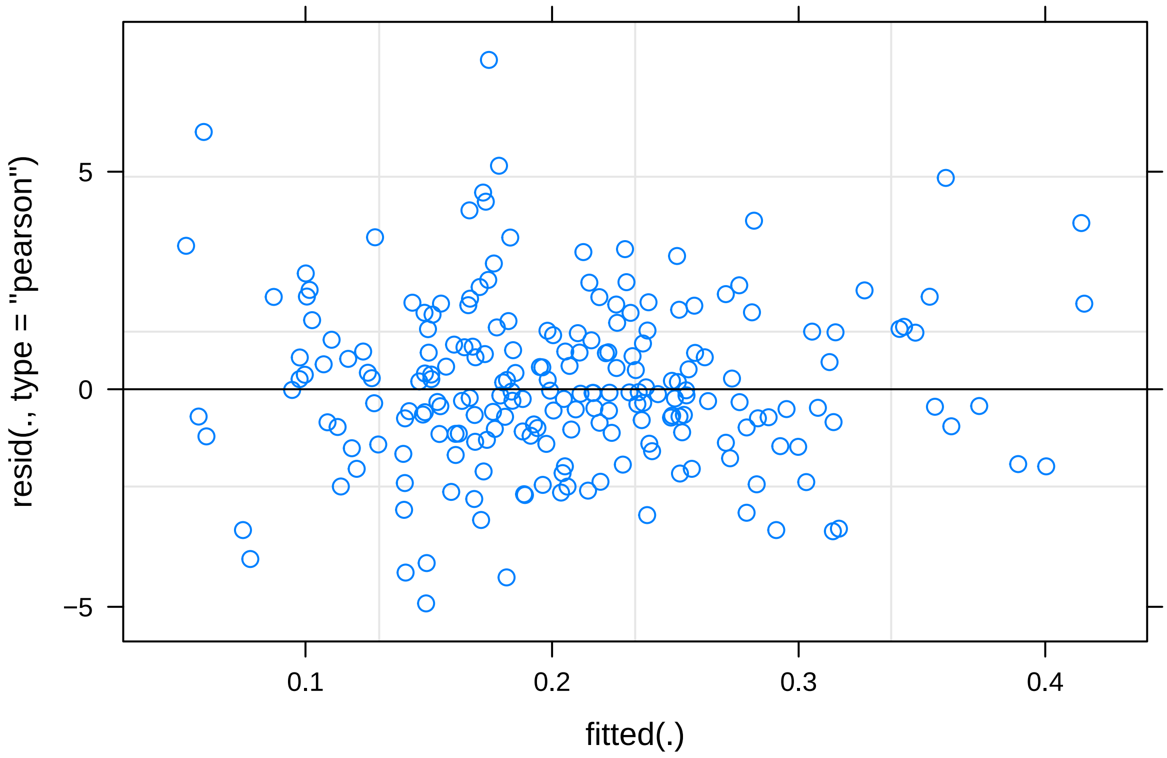
Question 5: Are birds more site faithful during the day or night?

```r
summary_df <- read.csv("F:/OneDrive - Liverpool John Moores University/Synced files/Projects/1_NIOZ pro
m_revisits1 <- glmer(cbind(visited2_cells, visited1_cells - visited2_cells) ~ tide_tod + offset(log(n_ti
summary(m_revisits1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(visited2_cells, visited1_cells - visited2_cells) ~ tide_tod +
```

```
##     offset(log(n_tides)) + (1 | year) + (1 | id)
##    Data: summary_df
##
##      AIC      BIC   logLik deviance df.resid
##   2540.8   2554.3  -1266.4   2532.8      212
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.9214 -0.9925 -0.0816  1.2921  7.5693
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.122973 0.35068
##  year   (Intercept) 0.006669 0.08166
## Number of obs: 216, groups:  id, 108; year, 5
##
## Fixed effects:
##               Estimate Std. Error z value            Pr(>|z|)
## (Intercept)   -3.86821    0.05248  -73.71 <0.0000000000000002 ***
## tide_todnight  0.39421    0.01581   24.94 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## tide_tdnght -0.101
```

```
plot(m_revisits1)
```




```
#plot(effect("tide_tod", m_revisits1))

newdf1 <- data.frame(revisits = 1,
                     summary_df[,c("tide_tod", "id", "year")],
                     n_tides = mean(summary_df$n_tides))
newdf1$revisits_prob <- predict(m_revisits1,newdata=newdf1,type="response")

summary_df$revisits_pc <- summary_df$visited2_cells/summary_df$visited1_cells

p5 <- ggplot()+
    geom_boxplot(data= newdf1, aes(x= tide_tod, y = revisits_prob, fill = tide_tod), outlier.shape= 4, c
    #geom_point(data= summary_df, aes(x= tide_tod, y = revisits_pc, fill = tide_tod), alpha=0.9, col="b
    scale_fill_manual(values=c("orange","grey25"))+
    scale_y_continuous(limits = c(0,0.5), expand = c(0,0), name = "revisit probability")+
    scale_x_discrete(name = "time")+
    theme_classic()+
    theme(legend.position="none")

p5
```

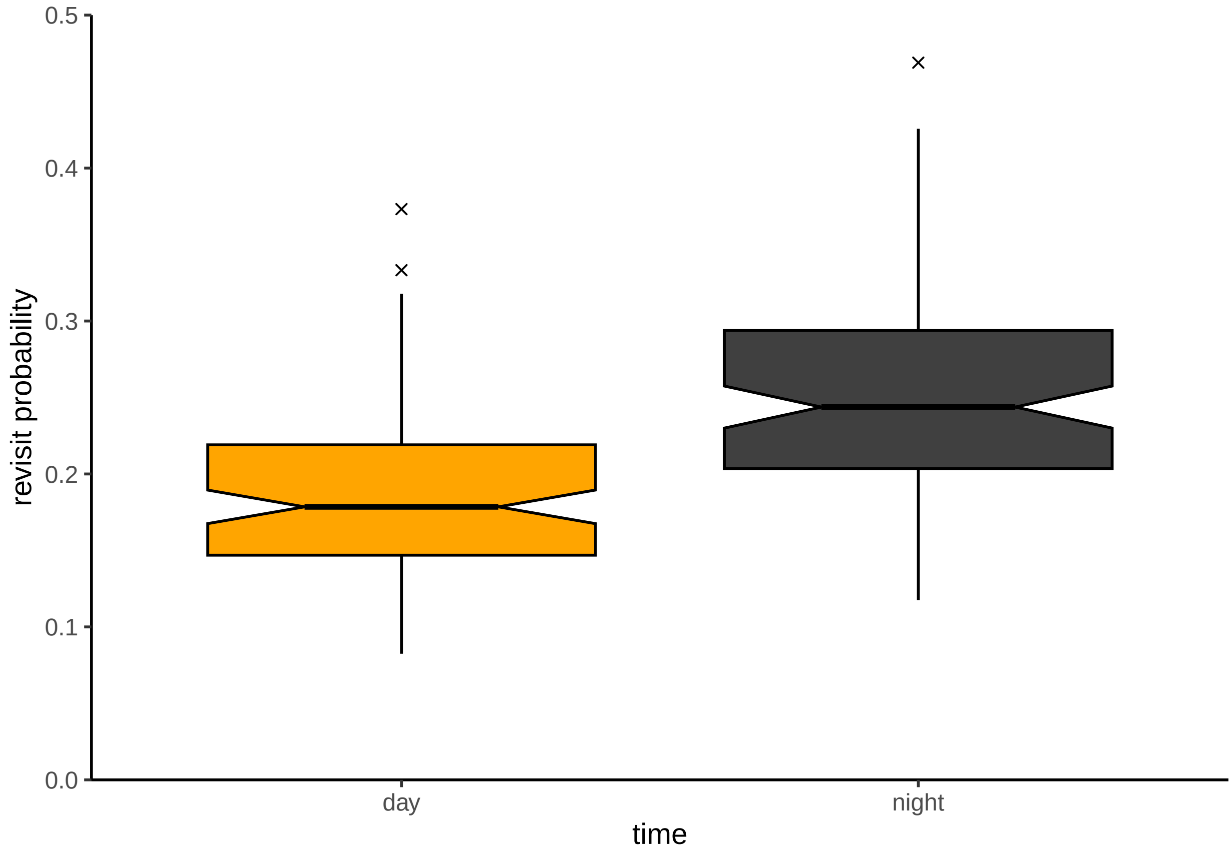
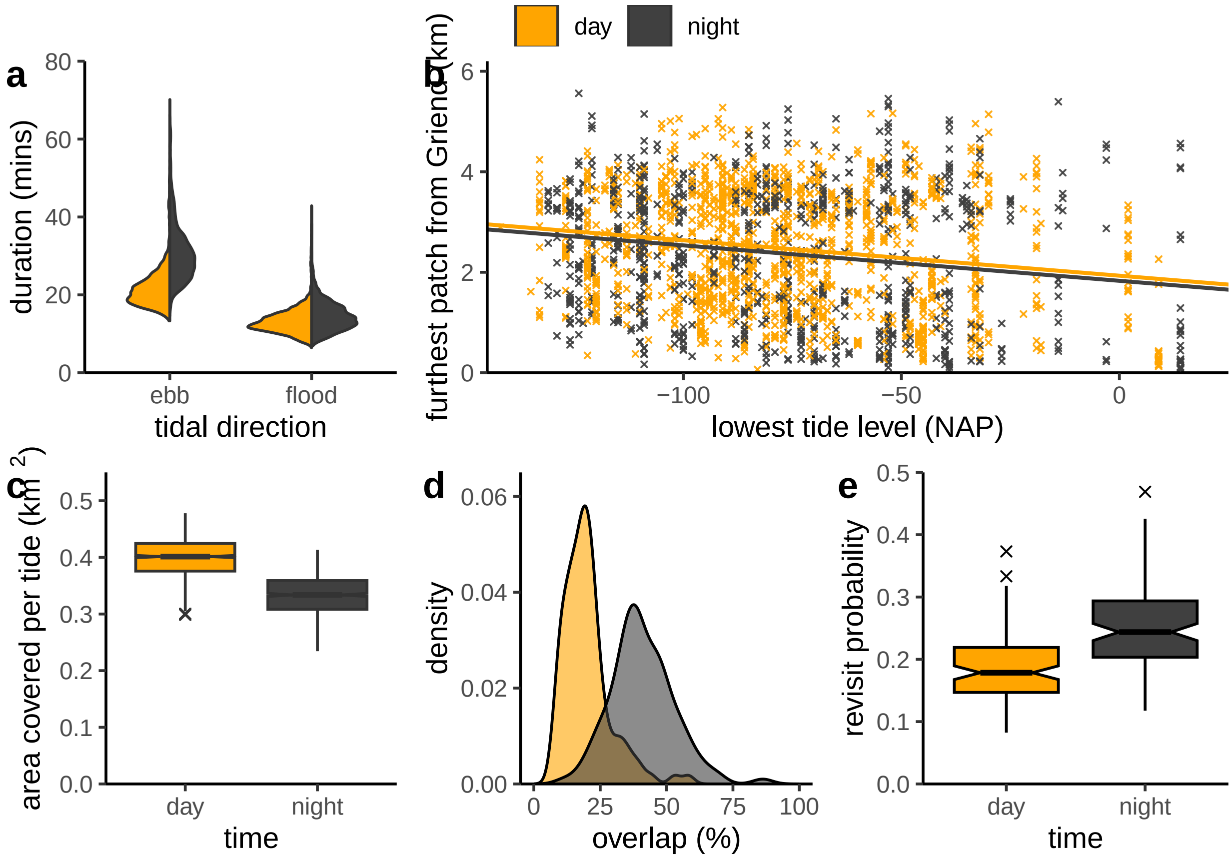


Fig. 2 merged plot

```
legend <- get_legend(p1 + theme(legend.position="top", legend.title = element_blank()))


r1 <- plot_grid(p1,p2, rel_widths = c(1,2), labels = c("a","b"))#, label_size=20)
r2<- plot_grid(p3,p4,p5, labels=c("c","d", "e"), nrow=1)#, label_size=20)
plot_all<- plot_grid(legend, r1, r2, nrow=3, rel_heights = c(0.1,1,1))

plot_all
```




```
#ggsave("F:\\OneDrive - Liverpool John Moores University\\Synced files\\Projects\\1_NIOZ projects\\2_Da
```

For supplementary fig S1

different thresholds for site fidelity

```
m_revisits2 <- glmer(cbind(visited3_cells, visited1_cells - visited3_cells) ~ tide_tod + offset(log(n_ti
m_revisits3 <- glmer(cbind(visited4_cells, visited1_cells - visited4_cells) ~ tide_tod + offset(log(n_ti
m_revisits4 <- glmer(cbind(visited5_cells, visited1_cells - visited5_cells) ~ tide_tod + offset(log(n_ti



newdf2 <- data.frame(revisits = 2,
                     summary_df[,c("tide_tod", "id", "year")],
                     n_tides = mean(summary_df$n_tides))
newdf2$revisits_prob <- predict(m_revisits2,newdata=newdf2,type="response")

newdf3 <- data.frame(revisits = 3,
                     summary_df[,c("tide_tod", "id", "year")],
                     n_tides = mean(summary_df$n_tides))
newdf3$revisits_prob <- predict(m_revisits3,newdata=newdf3,type="response")

newdf4 <- data.frame(revisits = 4,
```

```
                          summary_df[,c("tide_tod", "id", "year")],
                          n_tides = mean(summary_df$n_tides))
newdf4$revisits_prob <- predict(m_revisits4,newdata=newdf4,type="response")


newdf <- rbind(newdf1, newdf2, newdf3,newdf4)


realdf <- rbind(data.frame(revisits = 1, summary_df[,c("tide_tod", "id", "year")],  revisits_pc = summa
                data.frame(revisits = 2,summary_df[,c("tide_tod", "id", "year")],  revisits_pc = summary
                data.frame(revisits = 3, summary_df[,c("tide_tod", "id", "year")],  revisits_pc = summa
                data.frame(revisits = 4, summary_df[,c("tide_tod", "id", "year")],  revisits_pc = summa

newdf2 <- merge(newdf, realdf, by = c("revisits", "tide_tod", "id", "year"))
newdf2$revisits = factor(newdf$revisits)

ggplot()+
    geom_boxplot(data= newdf2, aes(x= revisits, y = revisits_prob, fill = tide_tod), col="black", notch=
    scale_fill_manual(values=c("orange","grey15"))+
    scale_color_manual(values=c("black", "black"))+
    facet_grid(cols = vars(year))+
    theme_bw()
```