

Activity Pattern Analysis

Nate Wehr

October 10, 2019

#Installing Packages

The first step of any R code is to make sure that the packages you need are installed. If the code below fails, it's because you haven't installed those packages. They can be installed in the console using the `< install.packages >` command.

#Importing Data

Prior to reading in the csv files, you need to do some basic data cleaning based-on suggestions from the literature. This consists of getting rid of non-independent detections (within 30 min) (Gerber et al. 2012; < <https://academic.oup.com/jmammal/article/93/3/667/835578> (<https://academic.oup.com/jmammal/article/93/3/667/835578>) >).

```
#import datasets to R
Initialfox = read.csv("Sample Fox - NRES 476 - Data.csv")
Initialhare = read.csv("Sample Hare - NRES 476 - Data.csv")

#displays first 6 rows of data showing that you have the right data
head(Initialfox)
```

##	Camera	Date	Time	Temp	Number	Species	Habitat	Trail	Attractants
## 1	SC1	11/22/2014	10:55:20	0	1	Red Fox	Forest	Manmade	Cover
## 2	SC1	1/27/2015	1:13:34	-9	1	Red Fox	Forest	Manmade	Cover
## 3	SC1	1/27/2015	2:11:37	-10	1	Red Fox	Forest	Manmade	Cover
## 4	SC1	2/12/2015	14:26:13	-20	1	Red Fox	Forest	Manmade	Cover
## 5	SC1	2/15/2015	13:27:41	-22	1	Red Fox	Forest	Manmade	Cover
## 6	SC1	3/12/2015	2:43:08	-11	1	Red Fox	Forest	Manmade	Cover

```
head(Initialhare)
```

##	Camera	Date	Time	Temp	Number	Species	Habitat	Trail
## 1	SC1	3/19/2015	0:29:07	-11	1	Snowshoe Hare	Forest	None
## 2	SC1	4/27/2015	23:32:17	0	1	Snowshoe Hare	Forest	None
## 3	SC1	4/28/2015	1:50:44	0	1	Snowshoe Hare	Forest	None
## 4	SC1	5/7/2015	0:43:55	7	1	Snowshoe Hare	Forest	None
## 5	SC1	5/7/2015	23:09:23	13	1	Snowshoe Hare	Forest	None
## 6	SC2	5/23/2015	2:50:14	0	1	Snowshoe Hare	Grassland	Small

##	Attractants
## 1	Cover
## 2	Cover
## 3	Cover
## 4	Cover
## 5	Cover
## 6	None

```
#make time cyclical
hare = mutate(Initialhare, fractime = hms(Time) / hms("24:00:00"), RadTime = fractime*2*pi)
fox = mutate(Initialfox, fractime = hms(Time) / hms("24:00:00"), RadTime = fractime*2*pi)
```

#Analyzing Data

The data above can be subsetted as much as necessary/desired. Of course subsetting will result in smaller sample sizes and higher variance around estimates.

The workhorse package that we'll be using to do these analyses is the package `< overlap >`. There's a very good tutorial on how to use the overlap package at < <https://cran.r-project.org/web/packages/overlap/vignettes/overlap.pdf> (<https://cran.r-project.org/web/packages/overlap/vignettes/overlap.pdf>) >.

Most of the methods we're using are based on a paper by Ridout and Linkie (2009) at < https://www.jstor.org/stable/pdf/20696577.pdf?casa_token=s005-yjLq8wAAAAA:rt2w90i12IzmlyPbuG0q2AeXYysi2VzacwjLiz0C3S5jfAm74y6bKMsDjg0A0zwwgqLeEudf4DgN9Wzl7Hy9isZ5AgTwwMquoTWqsT7K205RDwR >, which was adapted for movement data by Lashley et al. (2018) at < <https://www.nature.com/articles/s41598-018-22638-6> (<https://www.nature.com/articles/s41598-018-22638-6>) >.

casa_token=s005-

yjLq8wAAAAA:rt2w90i12IzmlyPbuG0q2AeXYysi2VzacwjLiz0C3S5jfAm74y6bKMsDjg0A0zwwgqLeEudf4DgN9Wzl7Hy9isZ5AgTwwMquoTWqsT7K205RDwR

([https://www.jstor.org/stable/pdf/20696577.pdf?casa_token=s005-](https://www.jstor.org/stable/pdf/20696577.pdf?casa_token=s005-yjLq8wAAAAA:rt2w90i12IzmlyPbuG0q2AeXYysi2VzacwjLiz0C3S5jfAm74y6bKMsDjg0A0zwwgqLeEudf4DgN9Wzl7Hy9isZ5AgTwwMquoTWqsT7K205RDwR)

yjLq8wAAAAA:rt2w90i12IzmlyPbuG0q2AeXYysi2VzacwjLiz0C3S5jfAm74y6bKMsDjg0A0zwwgqLeEudf4DgN9Wzl7Hy9isZ5AgTwwMquoTWqsT7K205RDwR

>, which was adapted for movement data by Lashley et al. (2018) at < <https://www.nature.com/articles/s41598-018-22638-6> >.

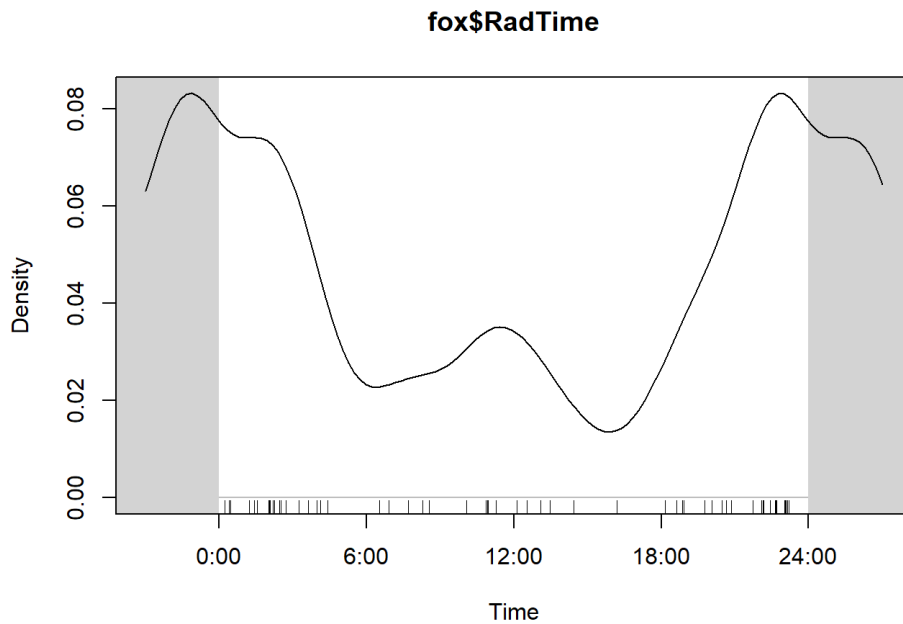
(<https://www.nature.com/articles/s41598-018-22638-6>) >.

An important note regarding sample size: a number of papers have been published with as few as 10 activity samples (e.g., independent photographs), but higher sample sizes are recommended. Lashley et al. (2018) found that activity patterns were closer to “truth” when at least 100 samples were used.

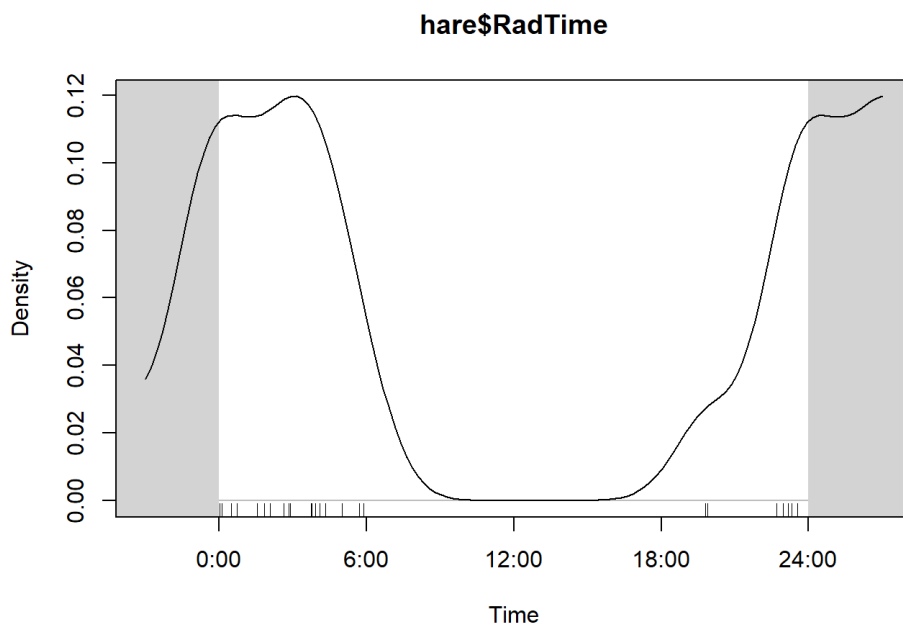
#Probability Density Functions

The next step is using the non-parametric kernel density approach (Ridout and Linkie 2009) to create probability density functions which can be used to evaluate during which part of the diel cycle a species is most active. These simply use the time of observations converted to a fraction of the diel cycle to display at which times the animal is most active.

```
#create a probability density plot showing animal activity over 24 hr period  
#< rug = TRUE > displays data points used to generate function across bottom of graph  
#< adjust = 0.8 > is the recommended setting for small sample sizes  
densityPlot(fox$RadTime, rug = TRUE, adjust=0.8)
```



```
densityPlot(hare$RadTime, rug = TRUE, adjust = 0.8)
```



As stated above, the adjust value determines the smoothing factor used to display the data. Ridout and Linkie (2009) recommend using an adjust value of 0.8 for small sample sizes based on simulations. We have small sample sizes for all of our species because we’re only using 20 cameras.

#Circular Statistics: Part 1

Because the diel cycle is circular (i.e. 23:59 one day becomes 00:00 the next day), it requires statistics that treat the data as such. We will be using packages `< circular >` and `< CircStats >`, and using the statistics suggested in Gerber et al. (2012) and Lashley et al. (2018).

First, we'll use Rao's test of uniformity to test whether the activity pattern of an animal is uniform over the diel cycle.

```
#transform data into circular class

Cfox = circular(fox$RadTime, units = "radians")
Chare = circular(hare$RadTime, units = "radians")

#statistical test for whether the data is uniformly distributed across time
rao.spacing.test(Cfox)
```

```
##
##      Rao's Spacing Test of Uniformity
##
## Test Statistic = 148.6842
## 0.05 < P-value < 0.10
##
```

```
rao.spacing.test(Chare)
```

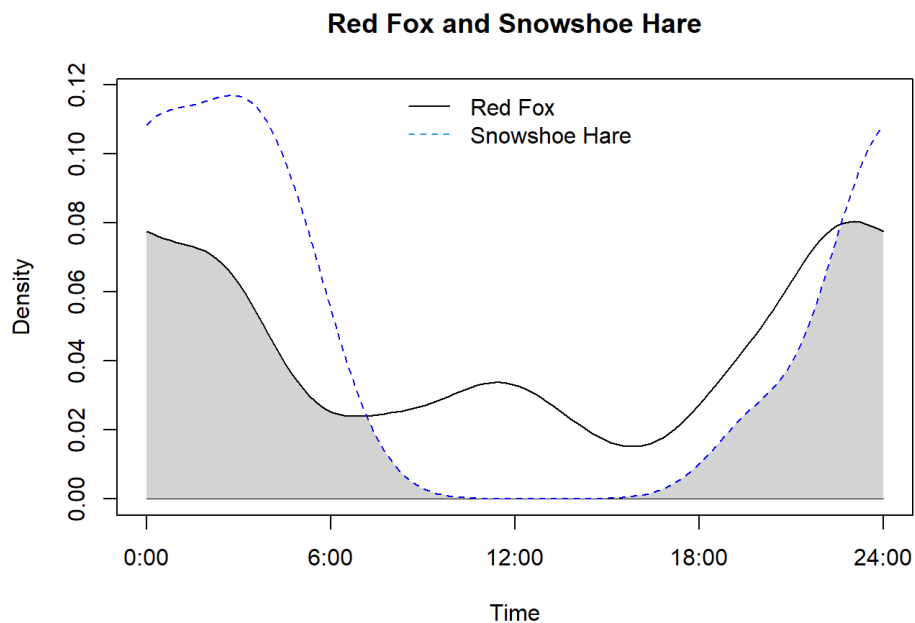
```
##
##      Rao's Spacing Test of Uniformity
##
## Test Statistic = 222.4542
## P-value < 0.001
##
```

From the stats above, we can conclude that the activity patterns of red fox weren't significantly different from a uniform distribution, while snowshoe hare did not have uniform activity patterns over the diel cycle.

#Estimating Temporal Overlap

Finally, we can evaluate the coefficient of temporal overlap (delta) between two probability density functions (e.g., species, seasons, years, etc.). There are a couple of different metrics for estimating overlap from Ridout and Linkie (2009). From their simulations they recommend `delta1` as performing best for data with limited sample sizes, and so that is what we will use. First, however, we will plot the overlap between the two species.

```
#plot temporal overlap
overlapPlot(fox$RadTime, hare$RadTime, main="Red Fox and Snowshoe Hare")
legend("top", c("Red Fox", "Snowshoe Hare"), lty=c(1,2), col = c(1,4), bty="n")
```



```
#estimates the coefficient of temporal overlap  
overlapEst(fox$RadTime, hare$RadTime, type = "Dhat1")
```

```
##      Dhat1  
## 0.6632196
```

#Circular Statistics: Part 2

To statistically test for a difference in the overlap, Lashley et al. (2018) use a Watson U2 test (also called a Watson two-sample test of heterogeneity), which determines whether two samples of circular data are homogenous. This can be used as support for conclusions along with estimates of temporal overlap.

```
#perform the watson U2 test  
watson.two(Cfox, Chare)
```

```
##  
##      Watson's Two-Sample Test of Homogeneity  
##  
## Test Statistic: 0.2761  
## 0.001 < P-value < 0.01  
##
```

From the stats above, we can conclude that the fox and the hare had significantly different activity patterns.