

ANALYZE ANIMAL BEHAVIOR DATA USING:

Animal

Matti Pastell

matti.pastell@helsinki.fi

Department of Agrotechnology &
Center of Animal Welfare, *University of Helsinki*

March 31, 2009

1 Introduction

The **Animal** package is a collection of functions for analyzing animal behavior data originating from a variety of sources. The package was originally created to analyze data files from CowLog (open source software for coding behaviors from digital video), but the functionality has been extended to cover also other data sources. This document briefly describes some the key features of the package via simple examples.

Animal has been created in the Research Centre of Animal Welfare ¹.

2 CowLog data

The package has basic analysis functions for analyzing time coded behavioral data coded with CowLog [1] ². The main function in the package is *cowAnalyze*. The function takes the data file name, labels of the codes and the type (event or state) of the codes as inputs, and gives a summary table and plot of the results as output. The function also removes double state (duplicated) errors for state codes.

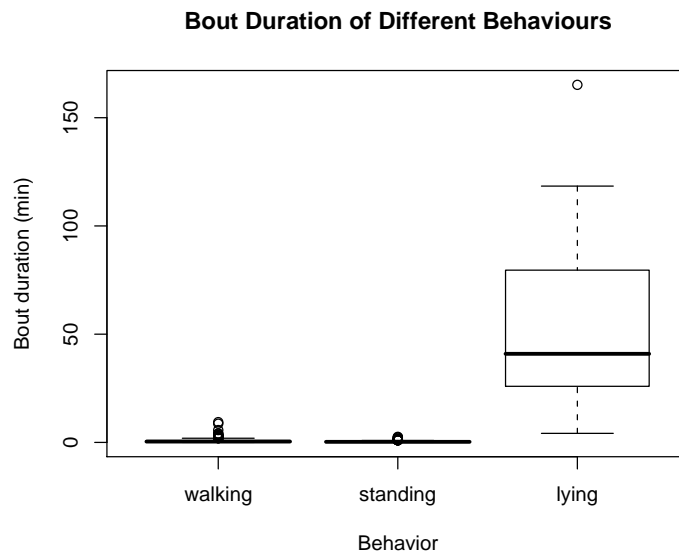
¹<http://http://www.vetmed.helsinki.fi/hyvinvointikeskus/english/index.htm>

²<http://www.mm.helsinki.fi/~mpastell/CowLog>

Here is a short example on how to use the function in R: Analyze CowLog datafile named calf1.bh1: Define codes 1-3 and give them names walking, standing and lying. The descriptive statistics will appear on screen and they are also saved to variable analyzed.

```
> library(Animal)
> analyzed <- cowAnalyze(file = "calf1.bh1", states = c(1,
+ 2, 3), state.names = c("walking", "standing", "lying"))
```

```
Event results
Frequency      0 times
State results   walking standing lying
Frequency      270 269 21 times
Bout           0.8 0.37 54.98 minutes
Total          214.81 99.35 1154.62 minutes
```



3 RIC data

The Animal package has several functions for working with the data produced by Insentec RIC feed measurement system. The basic function `read.RIC` can be used to read in RIC log files and they can be processed using `clean.RIC` and `bouts.RIC` functions.

Example data has been read in by using `read.RIC` with option `clean=F`. It is included in the package as dataset `RIC`.

```
> data(RIC)
```

First we clean the data from zero rows and negative feed intakes, then we merge the feeding bouts that are less than 5 minutes apart.

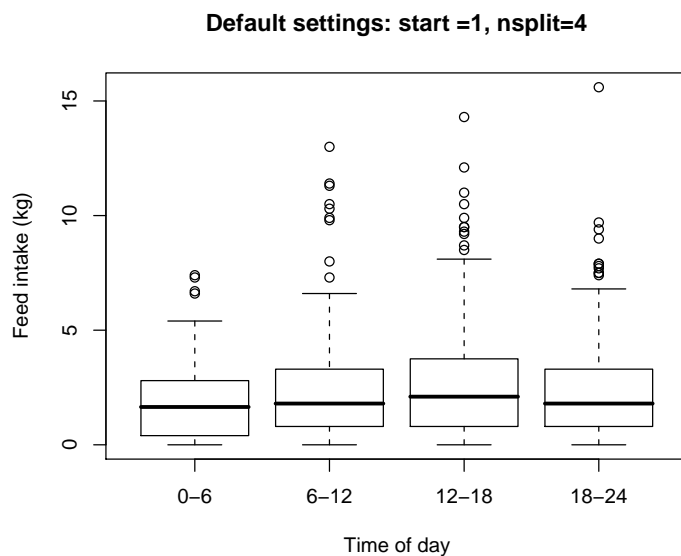
```
> RIC2 <- clean.RIC(RIC)
> bouts <- bouts.RIC(RIC2)
> head(bouts, 5)
```

	cowID	trough	begin	end	intake
1	320	11	2009-03-18 00:47:02	2009-03-18 01:12:03	4.8
2	320	11	2009-03-18 01:40:57	2009-03-18 01:51:53	2.6
3	320	11	2009-03-18 04:02:20	2009-03-18 04:22:30	2.9
4	320	11	2009-03-18 06:10:23	2009-03-18 07:24:27	9.8
5	320	11	2009-03-18 08:51:23	2009-03-18 08:56:11	1.0

	bout.duration	intake.duration
1	25.01667 mins	24.51667
2	10.93333 mins	10.93333
3	20.16667 mins	15.81667
4	74.06667 mins	63.13333
5	4.80000 mins	4.80000

We can plot the feed intake distribution during different hours of the day with the help of `partOfDay` function.

```
> boxplot(intake ~ partOfDay(begin), data = bouts, ylab = "Feed intake (kg)",
+         xlab = "Time of day", main = "Default settings: start =1, nsplit=4")
```



4 Summarizing time series

We are frequently interested in summarising events by an hour, day, week or a month. Say how many times a cow has visited the feeding trough during the day or what is the hourly sum of the feed intake for each animal. Animal provides functions to calculate these things easily with functions `hourly`, `daily`, `weekly` and `monthly`.³

The basic syntax for all functions is similar: We need to specify the data vector we want to analyze, the time stamps for the data in POSIXct format, the summarizing function and optionally the subject.

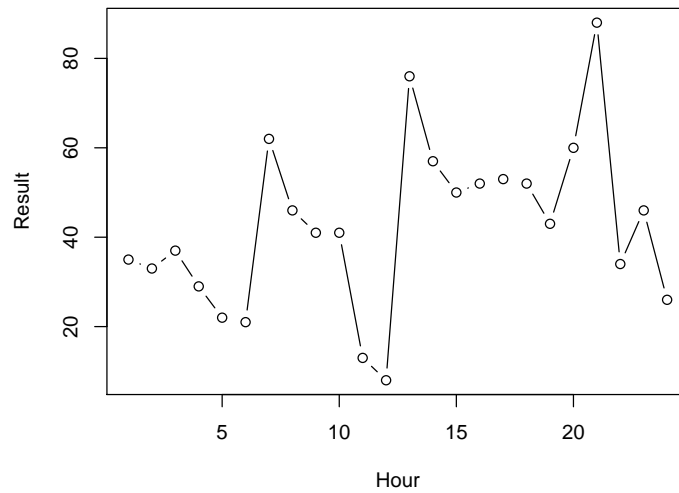
For instance, we are interested in what is the hourly (*summarized according to the start time of feeding event*) number of visits in the dataset `bouts` (created in RIC data chapter).

```
> attach(bouts)
> hourly.visits <- hourly(intake, begin, fun = length)
> head(hourly.visits)
```

	Hour	Result
1	1	35
2	2	33
3	3	37
4	4	29
5	5	22
6	6	21

```
> plot(Result ~ Hour, data = hourly.visits, type = "b")
```

³These are just simple wrapper functions to aggregate, but I feel these are easier to remember and more convenient in frequent use

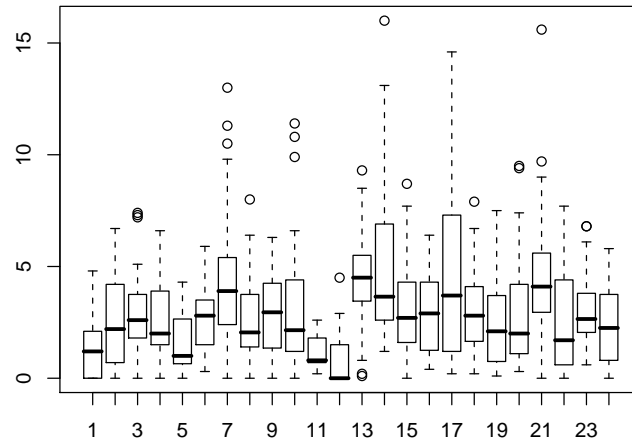


Similarly we can summarize the hourly intake for each cow:

```
> cow.intake <- hourly(intake, begin, fun = sum, subject = cowID)
> head(cow.intake)
```

	Hour	Subject	Result
1	1	320	4.8
2	2	320	2.6
3	5	320	2.9
4	7	320	9.8
5	9	320	1.0
6	10	320	3.8

```
> boxplot(Result ~ Hour, data = cow.intake)
> detach(bouts)
```



References

- [1] Hänninen, L. & Pastell, M. 2009. CowLog: Open source software for coding behaviors from digital video. Behavior Research Methods. 41(2), 472-476.