ChickenRanging NB2

Further data curation and creation of the files “hentimeslist” and “allhenstimestats”

Version: 1.0
Author: Bernhard Voelkl
Created in: *Mathematica* 10.2
Last modified: 15/02/2018

Initialization

```
SetDirectory["F://Chicken Research/ChickenRanging4"]
```

```
F:\Chicken Research\ChickenRanging4
```

```
data = Get["transitiondata"];
```

```
allhens = Get["allhens"];
```

```
columns = {"tag", "hen", "ranger", "pen", "date", "absolut time", "from", "to",
   "time", "WG open", "LH open", "FR open", "WG closed", "LH closed", "FR closed"}
```

```
{tag, hen, ranger, pen, date, absolut time, from, to,
 time, WG open, LH open, FR open, WG closed, LH closed, FR closed}
```

```
datelist = Union[data[[All, 5]]];
```

```
tags = allhens[[All, 3]];
```

Making lists with sojourn times “hentimeslist” and “allhentimestats”

Here I am creating two new files: “hentimeslist” and “allhentimestats”.
“hentimeslist” will be a very long list, where each line summarizes the sojourn of one hen in an area. So it will give the area, the time the animal went there and how long it stayed there (together with some other summaries for that henday). Sojourn times are only calculated for those instances where we have no missing antenna readings in between: so in line t in the transition file for that henday there must be the same area named in “to” as in line t+1 in “from”. If this is not the case, then no sojourn time is calculated. This means “hentimeslist” only contains those instances and times where we are sure where the hen is. (In this respect the data here differ from the data-files created for the time-warping: there it was necessary to have a continuous data stream and therefore we decided there to interpolate missing information. Here we don’t.)
“allhentimestats” is another list with one line per henday. It will summarize all observations for one hen for one day: how often it was in different areas, how much time it spent there, how many

'correct' and 'incorrect' transitions we observed (where 'correct' means a transition to an adjacent area) and what proportion of the time we can be sure about the animal's location.

```
ifnonzero[list_] := If[Length[list] > 0, Total[list[[All, 9]]]]
```

```
allhentimestats = {};
hentimeslist = {};
For[d = 1, d ≤ Length[datelist], d++,
            shortdata = Cases[data, {_, _, _, _, datelist[[d]], __}];
            For[t = 1, t ≤ Length[tags], t++,
    henday = Cases[shortdata, {tags[[t]], __}];
   If[Length[henday] > 1,
    hentimes = {};
    For[i = 1, i < Length[henday], i++,
     AppendTo[hentimes,
      If[henday[[i, 8]] == henday[[i + 1, 7]],
       Join[henday[[i + 1, 1 ;; 6]],
        {henday[[i, 9]], henday[[i + 1, 9]], henday[[i + 1, 9]] - henday[[i, 9]], henday[[i + 1, 7]],
         henday[[i, 10]],
         henday[[i, 13]],
         henday[[i, 13]] - henday[[i, 10]]
         }],
         {}
         ]
         ];
    validhentimes = DeleteCases[hentimes, {}];
    (* first add IN at start and end then calculate times ...  *)
    If[henday[[1, 7]] == "IN", validhentimes = Prepend[validhentimes,
       Join[henday[[1, 1 ;; 6]], {henday[[1, 10]], henday[[1, 9]], henday[[1, 9]] - henday[[1, 10]],
         "IN", henday[[1, 10]], henday[[1, 13]], henday[[1, 13]] - henday[[1, 10]]}]]
    ];
    If[henday[[-1, 8]] == "IN",
     validhentimes = Append[validhentimes,
       Join[henday[[1, 1 ;; 5]], {Round[henday[[-1, 6]] + henday[[-1, 9]]]},
        {henday[[-1, 9]], henday[[1, 13]], henday[[1, 13]] - henday[[-1, 9]], "IN",
         henday[[1, 10]], henday[[1, 13]], henday[[1, 13]] - henday[[1, 10]]}]]
    ];

                            totalobservedtime =
     If[Length[validhentimes] > 0, Total[validhentimes[[All, 9]]], 0.];
                            hentimestats = Join[henday[[1, 1 ;; 6]],
      {Length[hentimes], Count[hentimes, x_ /; x ≠ {}],
       Count[hentimes, x_ /; x == {}], totalobservedtime,
                                  If[(henday[[1, 13]] - henday[[1, 10]]) > 0,
        totalobservedtime / (henday[[1, 13]] - henday[[1, 10]]), "NA"],
       Count[validhentimes[[All, 10]], "IN"],
                                     ifnonzero[
        Cases[validhentimes, {__, "IN", __}]],
                                     Count[validhentimes[[All, 10]], "WG"],
       ifnonzero[Cases[validhentimes, {__, "WG", __}]],
                                     Count[validhentimes[[All, 10]], "LH"],
       ifnonzero[Cases[validhentimes, {__, "LH", __}]],
```

```
                                    Count[validhentimes[[All, 10]], "FR"],
    ifnonzero[Cases[validhentimes, {__, "FR", __}]]
                                  }
                              ];
                             AppendTo[allhentimestats, hentimestats];
                             If[Length[validhentimes] > 0,

 AppendTo[hentimeslist, Transpose[Append[Transpose[validhentimes],
    Table[totalobservedtime, {Length[validhentimes]}]]]]
                              ],
(*This block is for birds not observed on that day*)
validhentimes = {};
inpen = ToExpression[Cases[allhens, {_, _, tags[[t]]}][[1, 2]]];
hd = Cases[shortdata, {_, _, _, inpen, __}];
If[Length[hd] > 0,
 hd = hd[[1]];
 focushen = Cases[allhens, {_, _, tags[[t]]}][[1]];
 validhentimes =
  Prepend[validhentimes, Join[{focushen[[3]], focushen[[1]], "NA"}, hd[[4 ;; 6]],
    {hd[[10]], hd[[13]], hd[[13]] - hd[[10]], "IN", hd[[10]], hd[[13]], hd[[13]] - hd[[10]]}]];
 totalobservedtime = If[Length[validhentimes] > 0,
   Total[validhentimes[[All, 9]]], 0.];
                            hentimestats = Join[{focushen[[3]], focushen[[1]], "NA"},
   hd[[4 ;; 6]], {1, 1, 0, totalobservedtime,
                                  If[(hd[[13]] - hd[[10]]) > 0, totalobservedtime /
      (hd[[13]] - hd[[10]]), "NA"], Count[validhentimes[[All, 10]], "IN"],
                                    ifnonzero[
     Cases[validhentimes, {__, "IN", __}]],
                                    Count[validhentimes[[All, 10]], "WG"],
    ifnonzero[Cases[validhentimes, {__, "WG", __}]],
                                    Count[validhentimes[[All, 10]], "LH"],
    ifnonzero[Cases[validhentimes, {__, "LH", __}]],
                                    Count[validhentimes[[All, 10]], "FR"],
    ifnonzero[Cases[validhentimes, {__, "FR", __}]]
                                  }
                             ];
                            AppendTo[allhentimestats, hentimestats];
                            If[Length[validhentimes] > 0,

  AppendTo[hentimeslist, Transpose[Append[Transpose[validhentimes],
     Table[totalobservedtime, {Length[validhentimes]}]]]]
                             ]

                 ],
validhentimes = {};
inpen = ToExpression[Cases[allhens, {_, _, tags[[t]]}][[1, 2]]];
hd = Cases[shortdata, {_, _, _, inpen, __}];
If[Length[hd] > 0,
 hd = hd[[1]];
 focushen = Cases[allhens, {_, _, tags[[t]]}][[1]];
 validhentimes =
  Prepend[validhentimes, Join[{focushen[[3]], focushen[[1]], "NA"}, hd[[4 ;; 6]],
    {hd[[10]], hd[[13]], hd[[13]] - hd[[10]], "IN", hd[[10]], hd[[13]], hd[[13]] - hd[[10]]}]];
 totalobservedtime = If[Length[validhentimes] > 0,
```

```
        Total[validhentimes[[All, 9]], 0.];
                              hentimestats = Join[{focushen[[3]], focushen[[1]], "NA"},
        hd[[4 ;; 6]], {1, 1, 0, totalobservedtime,
                                    If[(hd[[13]] - hd[[10]]) > 0, totalobservedtime /
           (hd[[13]] - hd[[10]]), "NA"], Count[validhentimes[[All, 10]], "IN"],
                                      ifnonzero[
            Cases[validhentimes, {__, "IN", __}]],
                                      Count[validhentimes[[All, 10]], "WG"],
         ifnonzero[Cases[validhentimes, {__, "WG", __}]],
                                      Count[validhentimes[[All, 10]], "LH"],
         ifnonzero[Cases[validhentimes, {__, "LH", __}]],
                                      Count[validhentimes[[All, 10]], "FR"],
         ifnonzero[Cases[validhentimes, {__, "FR", __}]]
                                    }
                                ];
                              AppendTo[allhentimestats, hentimestats];
                              If[Length[validhentimes] > 0,

      AppendTo[hentimeslist, Transpose[Append[Transpose[validhentimes],
         Table[totalobservedtime, {Length[validhentimes]}]]]]
                                ]
    ]
   ](*End If Length henday>1*)

          ]
 ];(*End For d=1 to henday*)


allhentimestats = allhentimestats //. Null → 0.;

allhentimestats = Replace[allhentimestats, "" → "NA", {2}];

hentimeslist = Replace[hentimeslist, "" → "NA", {2}];

hentimeslist[[1, 1 ;; 3]]
```

```
{{DA09C002, 1, R, 11, Friday 02/09/2016, 3 681 795 768,
  32 400., 32 568., 168., IN, 32 400., 60 300., 27 900., 20 774.},
 {DA09C002, 1, R, 11, Friday 02/09/2016, 3 681 796 257, 33 056., 33 057., 1., FR,
  32 700., 60 300., 27 900., 20 774.}, {DA09C002, 1, R, 11, Friday 02/09/2016,
  3 681 796 488, 33 057., 33 288., 231., LH, 32 700., 60 300., 27 900., 20 774.}}
```

These are a few example lines for the file “hentimes”. Each line is one instance of a single area-visit by one hen (where we have observed both the transition into the area and the transition out). Columns are: RFID-tag, henID number, ranging category, pen number (11-14), date, absolute time of the start of the bout, time of the start of the bout (in seconds from 0:00:00), time of the end of the bout (in seconds from 0:00:00), duration of the bout (in seconds), area (“IN”: in the barn, “WG”: wintergarden, “LH”: yard (Laufhof), “FR”:free-range), opening time of WG on that day, closing time of WG on that day, opening duration WG on that day, time covered on that day for that bird (in seconds)}

```
allhentimestats[[100 ;; 103]]
```

```
{{DA7DAF52, 107, R, 14, Friday 02/09/2016,
  3 681 796 597, 5, 5, 0, 27 300., 1., 4, 23 421., 3, 3879., 0, 0., 0, 0.},
 {DA09B6D9, 108, NR, 14, Friday 02/09/2016, 3 681 795 791, 73, 68, 5,
  26 568., 0.973187, 28, 12 314., 34, 13 897., 8, 357., 0, 0.},
 {DA76015D, 109, R, 14, Friday 02/09/2016, 3 681 795 802, 58, 43, 15,
  19 836., 0.726593, 8, 10 421., 17, 8022., 17, 721., 3, 672.},
 {DA09BEB3, 110, R, 14, Friday 02/09/2016, 3 681 795 791, 17, 17,
  0, 27 300., 1., 10, 15 179., 9, 12 121., 0, 0., 0, 0.}}
```

These are a few example lines for the file allhentimestats. Each line summarizes the activities of one chicken on one day. Columns are:RFID-tag, henID number, ranging category, pen number (11-14), date, absoultetime of the first transition (by that bird on that day), Number of transitions (on that day for that hen), N correct transitions (where correct means that the bird is leaving an area it has previously moved into), Number of ambiguous transitions, total time observed (in seconds), proportion of total time (from WG open to WG closed) observed, Nunber of times observed in “IN”, total time (in seconds) spent in “IN”, Number of times observed in “WG”, total time (in seconds) in “WG”, Number of times observed in “LH”, total time (in seconds) spend in “LH”, Number of times observed in “FR”, total time (in seconds) spent in “FR”.

Remove early entries from birds that got transponder only later:

```
allhentimestats = DeleteCases[allhentimestats,
   {"DA760134", _, _, _, _, x_, __} /; x <= AbsoluteTime[{2016, 6, 16, 0, 0, 0}]];

allhentimestats = DeleteCases[allhentimestats,
   {"DA09BFC9", _, _, _, _, x_, __} /; x <= AbsoluteTime[{2016, 6, 16, 0, 0, 0}]];

earlydeath = {
   {"DA09BEC9", {2016, 9, 19, 0, 0, 0}},
   {"DA7DAF70", {2016, 7, 17, 0, 0, 0}},
   {"DA09BF80", {2016, 6, 18, 0, 0, 0}},
   {"DA7600A7", {2016, 6, 21, 0, 0, 0}},
   {"DA76019B", {2016, 6, 21, 0, 0, 0}},
   {"DA760094", {2016, 6, 23, 0, 0, 0}},
   {"DA09BA9B", {2016, 6, 23, 0, 0, 0}},
   {"DA09BED3", {2016, 8, 28, 0, 0, 0}},
   {"DA09BF02", {2016, 6, 14, 0, 0, 0}},
   {"DA09BEF7", {2016, 6, 18, 0, 0, 0}},
   {"DA09BEAD", {2016, 6, 22, 0, 0, 0}},
   {"DA09B72F", {2016, 6, 18, 0, 0, 0}},
   {"DA09B71A", {2016, 7, 1, 0, 0, 0}},
   {"DA09BA02", {2016, 7, 14, 0, 0, 0}},
   {"DA76014A", {2016, 6, 14, 0, 0, 0}},
   {"DA7600FB", {2016, 7, 17, 0, 0, 0}},
   {"DA09B747", {2016, 6, 23, 0, 0, 0}},
   {"DA09BF53", {2016, 6, 28, 0, 0, 0}},
   {"DA09B7C3", {2016, 7, 1, 0, 0, 0}},
   {"DA7DAF4C", {2016, 6, 23, 0, 0, 0}}
  };
```

```
For[i = 1, i ≤ Length[earlydeath], i++,
  allhentimestats = DeleteCases[allhentimestats,
    {earlydeath[[i, 1]], _, _, _, _, x_, __} /; x >= AbsoluteTime[earlydeath[[i, 2]]]]
 ];

hentimeslist = DeleteCases[hentimeslist,
   {{"DA760134", _, _, _, _, x_, __}, ___} /; x <= AbsoluteTime[{2016, 6, 16, 0, 0, 0}]];

hentimeslist = DeleteCases[hentimeslist,
   {{"DA09BFC9", _, _, _, _, x_, __}, ___} /; x <= AbsoluteTime[{2016, 6, 16, 0, 0, 0}]];

For[i = 1, i ≤ Length[earlydeath], i++,
 hentimeslist = DeleteCases[hentimeslist,
   {{earlydeath[[i, 1]], _, _, _, _, x_, __}, ___} /; x >= AbsoluteTime[earlydeath[[i, 2]]]]
]
```

Adding Order going out and back in

Here two more columns are added to the file “allhentimestats”: the first is the Order number in which the hen first appeared in the LH on that day (i.e. 1 means that out of the about 120 hens of the respective pen the focus hen was the first to appear in the LH on that day; 2 the second, etc..; “NA” means that on that day this hen did not go into the LH at all). the second is the Order number in which the hen last left the LH (going int WG) on that day (i.e. 1 means that out of the about 120 hens of the respective pen the focus hen was the first to finally return on that day; 2 the second, etc..; “NA” means that on that day this hen did not go into the LH at all.)

```
getfirstLH[pen_, date_] := Module[{firstlaufhof = {}, tolh, list},
  list = Cases[hentimeslist, {{_, _, _, pen, date, __}, __}];
  For[i = 1, i ≤ Length[list], i++,
   tolh = Cases[list[[i]], {__, "LH", __}];
   If[Length[tolh] > 0, AppendTo[firstlaufhof, tolh[[1, {7, 1}]]]
  ];
  Transpose[
   Join[{Table[pen, {Length[firstlaufhof]}], Table[date, {Length[firstlaufhof]}]},
    Transpose[Sort[firstlaufhof]], {Range[Length[firstlaufhof]]}]]
 ]

firstlh = Flatten[
   Table[Table[getfirstLH[p, datelist[[d]]], {p, 11, 14}], {d, 1, Length[datelist]}], 2];
```

... Transpose: The first two levels of {} cannot be transposed.

... Join: Heads List and Transpose at positions 1 and 2 are expected to be the same.

```
addorder[line_] := Module[{first, newline},
  first = Cases[firstlh, {_, line[[5]], _, line[[1]], _}];
  newline = If[Length[first] > 0, Append[line, first[[1, -1]]], Append[line, "NA"]];
  newline
 ]

allhentimestats = Table[addorder[allhentimestats[[i]]], {i, Length[allhentimestats]}];
```

```
getlastLH[pen_, date_] := Module[{lastlaufhof = {}, towg, list},
  list = Cases[hentimeslist, {{_, _, _, pen, date, __}, __}];
  For[i = 1, i ≤ Length[list], i++,
   towg = Cases[list[[i]], {__, "LH", __}];
   If[Length[towg] > 0, AppendTo[lastlaufhof, towg[[-1, {8, 1}]]]]
  ];
  Transpose[
   Join[{Table[pen, {Length[lastlaufhof]}], Table[date, {Length[lastlaufhof]}]},
    Transpose[Sort[lastlaufhof]], {Range[Length[lastlaufhof]]}]]
 ]

lastlh = Flatten[
   Table[Table[getlastLH[p, datelist[[d]], {p, 11, 14}], {d, 1, Length[datelist]}], 2];
```

... Transpose: The first two levels of {} cannot be transposed.

... Join: Heads List and Transpose at positions 1 and 2 are expected to be the same.

```
addorderin[line_] := Module[{first, newline},
  first = Cases[lastlh, {_, line[[5]], _, line[[1]], _}];
  newline = If[Length[first] > 0, Append[line, first[[1, -1]]], Append[line, "NA"]];
  newline
 ]

allhentimestats = Table[addorderin[allhentimestats[[i]], {i, Length[allhentimestats]}];
```

Adding Proportion spent “outdoors”

This adds a column to allhentimestats with the proportion of the observed time (from WG open to WG closed) the animal had spent in the outdoor area (LH and FR combined).

```
allhentimestats = Transpose[Append[Transpose[allhentimestats],
    (allhentimestats[[All, 17]] + allhentimestats[[All, 19]]) / allhentimestats[[All, 10]]]];
```

Save files

allhentimestats: Each line summarizes the activities of one chicken on one day. Columns are:RFID-tag, henID number, ranging category, pen number (11-14), date, absoultetime of the first transition (by that bird on that day), Number of transitions (on that day for that hen), N correct transitions (where correct means that the bird is leaving an area it has previously moved into), Number of ambiguous transitions, total time observed (in seconds), proportion of total time (from WG open to WG closed) observed, Nunber of times observed in “IN”, total time (in seconds) spent in “IN”, Number of times observed in “WG”, total time (in seconds) in “WG”, Number of times observed in “LH”, total time (in seconds) spend in “LH”, Number of times observed in “FR”, total time (in seconds) spent in “FR”, order appearing first in the LH (of tagged hens in the respective pen; based on start of first bout in LH), order of finally leaving the LH (of tagged hens in the respective pen; based on end of last bout in LH), the proportion of the observed time (from WG open to WG closed) the animal had spent in the outdoor area (LH and FR combined).

```
Save["allhentimestats", allhentimestats]
```

```
Export["allhentimestats.csv", allhentimestats]
```

allhentimestats.csv

hentimeslist: Each line is one instance of a single area-visit by one hen (where we have observed both the transition into the area and the transition out). Columns are: RFID-tag, henID number, ranging category, pen number (11-14), date, absolute time of the start of the bout, time of the start of the bout (in seconds from 0:00:00), time of the end of the bout (in seconds from 0:00:00), duration of the bout (in seconds), area (“IN”: in the barn, “WG”: wintergarden, “LH”: yard (Laufhof), “FR”:free-range), opening time of WG on that day, closing time of WG on that day, opening duration WG on that day, time covered on that day for that bird (in seconds)}

```
Save["hentimeslist", hentimeslist]
```

```
timematrix = Flatten[hentimeslist, 1];
```

timematrix.csv: cvs Matrix version of hentimeslist in Mathermatica. Each line is one instance of a single area-visit by one hen (where we have observed both the transition into the area and the transition out). Columns are: RFID-tag, henID number, ranging category, pen number (11-14), date, absolute time of the start of the bout, time of the start of the bout (in seconds from 0:00:00), time of the end of the bout (in seconds from 0:00:00), duration of the bout (in seconds), area (“IN”: in the barn, “WG”: wintergarden, “LH”: yard (Laufhof), “FR”:free-range), opening time of WG on that day, closing time of WG on that day, opening duration WG on that day, time covered on that day for that bird (in seconds)}

```
Export["timedata.csv", timematrix]
```

timedata.csv

```
(*-------------------------*)
```