

ChickenRanging NB1

Import of raw data and first data curation steps.

Creation of the files “transitiondata”, “allhens” and “henlinesX”

Version: 4.0.

Author: Bernhard Voelkl

Created in: *Mathematica* 10.2

Last modified: 01/03/2018

Initialization

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

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

Data Import

Basis for all further analyses are 5 raw data files:

Hen-list.xlsx (last modified: 2.6.2017) and Ranging-days.xlsx (last modified: 2.6.2017) from G:\VPHI\Welfare\2- Research Projects\Sabine Vögeli\Ranging\Auswertungen
INWGV2.csv (last modified: 23.5.2017), LHFRV2.csv (last modified: 8.5.2017) and WGLHV2.csv (last modified: 23.5.2017) from G:\VPHI\Welfare\2- Research Projects\Sabine Gebhardt\Ranging Study

Functions

```
secs[{h_, m_, s_}] := 3600 h + 60 m + s
```

```
makeLine[line_] := Join[line[[1 ;; 4]],  
  {If[line[[5]] == ".", ".", DateString[{line[[5]], {"Day", "Month", "Year"}},  
    {"DayName", " ", "Day", "/", "Month", "/", "Year"}]}],  
  If[line[[5]] == "." || line[[8]] == ".", ".", AbsoluteTime[  
    Join[DateList[{line[[5]], {"Day", "Month", "Year"}}][[1 ;; 3]], DateList[  
      {line[[8]], {"Hour", "Minute", "Second"}}][[4 ;; 6]]], line[[6]], line[[7]],  
  If[line[[8]] == ".", ".", secs[DateList[{line[[8]], {"Hour", "Minute", "Second"}}][[  
    4 ;; 6]]]}],  
  Table[If[line[[col]] == ".", ".", secs[DateList[{line[[col]], {"Hour", "Minute"}}][[  
    4 ;; 6]]], {col, 9, 14, 1}]  
]
```

Create new dataset

```
data = Import["LHFRV2.csv"];
```

```
data = Drop[data, 1];
```

```
newdata1 = ParallelMap[makeLine, data];
```

```
data = Import["WGLHV2.csv"];
```

```
data = Drop[data, 1];
```

```
newdata2 = ParallelMap[makeLine, data];
```

```
data = Import["INWGV2.csv"];
```

```
data = Drop[data, 1];
```

```
newdata3 = ParallelMap[makeLine, data];
```

```
newdata = Join[newdata1, newdata2, newdata3];
```

```
newdata = Sort[newdata, #1[[6]] < #2[[6]] &];
```

The 3 raw .csv files “LHFRV2”, “WGLHV2” and “INWGV2” with the antenna readings are imported, reformatted and merged into a single file “newdata”.

```
Dimensions[newdata]
```

```
{2157414, 15}
```

```
{2157414, 15}
```

This is the size of the merged data file. Each line is a single antenna reading. The 15 columns are given below: “tag”: Hex-number of the RFID tag, “hen”: unique hen ID number, “ranger”: categorization based on an initial rough classification (not further used), “pen”: pen ID number (11-14), “date”, “absolut time” (in sec), “from”: the area from which the hen is coming (IN: indoors (barn), WG: wintergarden, LH: yard (“Laufhof”), FR: free-range), “to”: the area where the hen is going to (same as before), “time”: the time of the day (in seconds counted from 0:00:00), “WG open”: the time (in seconds from 0:00:00) when the WG was opened on that day for that pen, “LH open”: the time (in seconds from 0:00:00) when the LH was opened on that day for that pen, “FR open”: the time (in seconds from 0:00:00) when the FR was opened on that day for that pen, “WG closed”: the time (in seconds from 0:00:00) when the WG was closed on that day for that pen, “LH closed”: the time (in seconds from 0:00:00) when the LH was closed on that day for that pen, “FR closed”: the time (in seconds from 0:00:00) when the FR was closed on that day for that pen

```
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}
```

```
newdata[[1200000]]
{DA760134, 934, NA, 12, Monday 26/09/2016, 3 683 872 615,
IN, WG, 35 815., 32 400., 32 400., 32 400., 58 800., 58 800., 58 800.}
```

this is an example line

Remove Entries without Date

```
Length[Cases[newdata, {_, _, _, _, ".", __}]]
82

newdata = DeleteCases[newdata, {_, _, _, _, ".", __}];
```

Restrict Date Range

```
end = AbsoluteTime[{2016, 10, 24, 0, 0, 0}];
(*delete those entries after October 24th*)

newdata = DeleteCases[newdata, {_, _, _, _, _, x_, __} /; x ≥ end];
```

```
rangingdays = Import["Ranging-days.xlsx"][[1]];
```

```
rangingdays[[1]]
{Day, StatusComm, EventComm, Pen1AntProb,
Pen2AntProb, Pen3AntProb, Pen4AntProb, DontUseDate}
```

There were a couple of days where there were some reported problems (see protocol ranging-days), where we decided to exclude those days.

```
daystoberemoved = Cases[rangingdays, {__, True}][[All, 1]]
{10.05.2016, 11.05.2016, 12.05.2016, 13.05.2016, 14.05.2016, 15.05.2016, 18.05.2016,
23.05.2016, 24.05.2016, 31.05.2016, 01.06.2016, 02.06.2016, 03.06.2016,
04.06.2016, 05.06.2016, 08.06.2016, 09.06.2016, 10.06.2016, 13.06.2016,
14.06.2016, 15.06.2016, 16.06.2016, 29.06.2016, 02.07.2016, 03.07.2016,
04.07.2016, 05.07.2016, 02.08.2016, 03.08.2016, 04.08.2016, 05.08.2016}

daystoberemoved = Table[DateString[{daystoberemoved[[i]], {"Day", "Month", "Year"}},
{"DayName", " ", "Day", "/", "Month", "/", "Year"}], {i, Length[daystoberemoved]}]
{Tuesday 10/05/2016, Wednesday 11/05/2016, Thursday 12/05/2016, Friday 13/05/2016,
Saturday 14/05/2016, Sunday 15/05/2016, Wednesday 18/05/2016, Monday 23/05/2016,
Tuesday 24/05/2016, Tuesday 31/05/2016, Wednesday 01/06/2016, Thursday 02/06/2016,
Friday 03/06/2016, Saturday 04/06/2016, Sunday 05/06/2016, Wednesday 08/06/2016,
Thursday 09/06/2016, Friday 10/06/2016, Monday 13/06/2016, Tuesday 14/06/2016,
Wednesday 15/06/2016, Thursday 16/06/2016, Wednesday 29/06/2016,
Saturday 02/07/2016, Sunday 03/07/2016, Monday 04/07/2016, Tuesday 05/07/2016,
Tuesday 02/08/2016, Wednesday 03/08/2016, Thursday 04/08/2016, Friday 05/08/2016}

Length[Cases[newdata, {_, _, _, _, x_, __} /; MemberQ[daystoberemoved, x]]]
60 831

newdata = DeleteCases[newdata, {_, _, _, _, x_, __} /; MemberQ[daystoberemoved, x]];
```

Also remove pen 11 October 10 th data (see protocol ranging-days):

```
newdata = DeleteCases[newdata,
  {_, _, _, 11, DateString[{"10.10.2016", {"Day", "Month", "Year"}]},
  {"DayName", " ", "Day", "/", "Month", "/", "Year"}], _];
```

Remove Tags

```
newdata = DeleteCases[newdata, {_, ".", _}];
```

Lines without a proper tag reading are removed

Remove Birds

A few birds died half-way during the experiment, so we decided to exclude them

```
henlist = Import["Hen-list.xlsx"][[1];
```

```
henlist[[1]]
```

```
{HenID, Pen, TagID, DeathDate, OrigRangClass, TrackComm, Use}
```

```
excludethosehens = Round[Cases[henlist, {_, False}]] [[All, 1]]
```

```
{5, 58, 77, 96, 116, 118, 514, 621, 639, 644,
 670, 678, 683, 701, 739, 757, 804, 857, 864, 866, 887, 925}
```

Create and export the file with all valid hens “allhens”

Columns are: HenID, Pen, TagID

```
allhens = Cases[henlist, {_, True}]] [[All, 1 ;; 3]];
```

```
allhens[[1]]
```

```
{1., 11, DA09C002}
```

```
roundallhens[{id_, pen_, tag_}] := {Round[id], pen, tag}
```

```
allhens = Map[roundallhens, allhens];
```

```
Save["allhens", allhens]
```

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

```
allhens.csv
```

Adjust Opening Times

Here comes a long and a bit clumsy routine for adjusting opening and closing times of the areas. Opening and closing times were recorded by the personnel usually to 5 min accuracy, however,

sometimes these times were just guesses or the watches of the caretaker was inaccurate (probably not a swiss watch then), so that we have sometimes already recordings in some areas even though they were only 'officially' opened later on. If this happens we re-adjust the opening times (in 5-minute intervals) so that the new opening time is before the first recording in an area. The same was done for the closing times --here time guesses were even less precise.

```

datelist = Union[DeleteCases[newdata, {_, "."}]][[All, 5]];

data = newdata;

newdata = {};
openingtimesordered = {};
alltimeshifts = {};
alldeletions = {};
timeorder = {};
For[d = 1, d ≤ Length[datelist], d++,
  For[p = 11, p ≤ 14, p++,
    penday = Select[data, #[[5]] == datelist[[d]] && #[[4]] == p &];
    penday = Sort[penday, #1[[6]] < #2[[6]] &];
    openingtimeWG = penday[[1, 10]];
    openingtimeLH = penday[[1, 11]];
    openingtimeFR = penday[[1, 12]];
    fiveminutesearlyWG = Cases[penday, {_, "WG", ___, x_, owg_,
      olh_, ofr_, cwg_, clh_, cfr_} /; x < owg && x ≥ owg - 300 && owg ≠ "."];
    tenminutesearlyWG = Cases[penday, {_, "WG", ___, x_, owg_, olh_, ofr_,
      cwg_, clh_, cfr_} /; x < owg - 300 && x ≥ owg - 600 && owg ≠ "."];
    If[Length[fiveminutesearlyWG] > 0, openingtimeWG = openingtimeWG - 300];
    If[Length[tenminutesearlyWG] > 0, openingtimeWG = openingtimeWG - 300];
    fiveminutesearlyLH = Cases[penday, {_, "LH", ___, x_, owg_,
      olh_, ofr_, cwg_, clh_, cfr_} /; x < olh && x ≥ olh - 300 && olh ≠ "."];
    tenminutesearlyLH = Cases[penday, {_, "LH", ___, x_, owg_, olh_, ofr_,
      cwg_, clh_, cfr_} /; x < olh - 300 && x ≥ olh - 600 && olh ≠ "."];
    If[Length[fiveminutesearlyLH] > 0, openingtimeLH = openingtimeLH - 300];
    If[Length[tenminutesearlyLH] > 0, openingtimeLH = openingtimeLH - 300];
    fiveminutesearlyFR = Cases[penday, {_, "FR", ___, x_, owg_,
      olh_, ofr_, cwg_, clh_, cfr_} /; x < ofr && x ≥ ofr - 300 && ofr ≠ "."];
    tenminutesearlyFR = Cases[penday, {_, "FR", ___, x_, owg_, olh_, ofr_,
      cwg_, clh_, cfr_} /; x < ofr - 300 && x ≥ ofr - 600 && ofr ≠ "."];
    If[Length[fiveminutesearlyFR] > 0, openingtimeFR = openingtimeFR - 300];
    If[Length[tenminutesearlyFR] > 0, openingtimeFR = openingtimeFR - 300];
    closingtimeWG = penday[[1, 13]];
    closingtimeLH = penday[[1, 14]];
    closingtimeFR = penday[[1, 15]];
    fiveminuteslateWG =
    Cases[penday, {_, "WG", ___, x_, owg_, olh_, ofr_, cwg_, clh_, cfr_} /;
      x > cwg && x ≤ cwg + 300 && cwg ≠ "."];
    tenminuteslateWG = Cases[penday, {_, "WG", ___, x_, owg_, olh_, ofr_,
      cwg_, clh_, cfr_} /; x > cwg + 300 && x ≤ cwg + 600 && cwg ≠ "."];
    fifteenminuteslateWG = Cases[penday, {_, "WG", ___, x_, owg_, olh_,
      ofr_, cwg_, clh_, cfr_} /; x > cwg + 600 && x ≤ cwg + 900 && cwg ≠ "."];
    twentyminuteslateWG = Cases[penday, {_, "WG", ___, x_, owg_, olh_,
      ofr_, cwg_, clh_, cfr_} /; x > cwg + 900 && x ≤ cwg + 1200 && cwg ≠ "."];
    twentyfiveminuteslateWG = Cases[penday, {_, "WG", ___, x_, owg_, olh_,
      ofr_, cwg_, clh_, cfr_} /; x > cwg + 1200 && x ≤ cwg + 1500 && cwg ≠ "."];
  ]
]

```

```

    thirtyminuteslateWG = Cases[penday, {_, "WG", ___, x_, owg_, olh_,
      ofr_, cwg_, clh_, cfr_} /; x > cwg + 1500 && x <= cwg + 1800 && cwg != "."];
  If[Length[fiveminuteslateWG] > 0, closingtimeWG = closingtimeWG + 300];
  If[Length[tenminuteslateWG] > 0, closingtimeWG = closingtimeWG + 300];
  If[Length[fifteenminuteslateWG] > 0, closingtimeWG = closingtimeWG + 300];
  If[Length[twentyminuteslateWG] > 0, closingtimeWG = closingtimeWG + 300];
  If[Length[twentyfiveminuteslateWG] > 0, closingtimeWG = closingtimeWG + 300];
  If[Length[thirtyminuteslateWG] > 0, closingtimeWG = closingtimeWG + 300];
  fiveminuteslateLH = Cases[penday, {_, "LH", ___, x_, owg_,
    olh_, ofr_, cwg_, clh_, cfr_} /; x > clh && x <= clh + 300 && clh != "."];
  tenminuteslateLH = Cases[penday, {_, "LH", ___, x_, owg_, olh_, ofr_,
    cwg_, clh_, cfr_} /; x > clh + 300 && x <= clh + 600 && clh != "."];
  fifteenminuteslateLH = Cases[penday, {_, "LH", ___, x_, owg_, olh_,
    ofr_, cwg_, clh_, cfr_} /; x > clh + 600 && x <= clh + 900 && clh != "."];
  twentyminuteslateLH = Cases[penday, {_, "LH", ___, x_, owg_, olh_,
    ofr_, cwg_, clh_, cfr_} /; x > clh + 900 && x <= clh + 1200 && clh != "."];
  twentyfiveminuteslateLH = Cases[penday, {_, "LH", ___, x_, owg_, olh_,
    ofr_, cwg_, clh_, cfr_} /; x > clh + 1200 && x <= clh + 1500 && clh != "."];
  thirtyminuteslateLH = Cases[penday, {_, "LH", ___, x_, owg_, olh_,
    ofr_, cwg_, clh_, cfr_} /; x > clh + 1500 && x <= clh + 1800 && clh != "."];
  If[Length[fiveminuteslateLH] > 0, closingtimeLH = closingtimeLH + 300];
  If[Length[tenminuteslateLH] > 0, closingtimeLH = closingtimeLH + 300];
  If[Length[fifteenminuteslateLH] > 0, closingtimeLH = closingtimeLH + 300];
  If[Length[twentyminuteslateLH] > 0, closingtimeLH = closingtimeLH + 300];
  If[Length[twentyfiveminuteslateLH] > 0, closingtimeLH = closingtimeLH + 300];
  If[Length[thirtyminuteslateLH] > 0, closingtimeLH = closingtimeLH + 300];
  fiveminuteslateFR = Cases[penday, {_, "FR", ___, x_, owg_,
    olh_, ofr_, cwg_, clh_, cfr_} /; x > cfr && x <= cfr + 300 && cfr != "."];
  tenminuteslateFR = Cases[penday, {_, "FR", ___, x_, owg_, olh_, ofr_,
    cwg_, clh_, cfr_} /; x > cfr + 300 && x <= cfr + 600 && cfr != "."];
  fifteenminuteslateFR = Cases[penday, {_, "FR", ___, x_, owg_, olh_,
    ofr_, cwg_, clh_, cfr_} /; x > cfr + 600 && x <= cfr + 900 && cfr != "."];
  twentyminuteslateFR = Cases[penday, {_, "FR", ___, x_, owg_, olh_,
    ofr_, cwg_, clh_, cfr_} /; x > cfr + 900 && x <= cfr + 1200 && cfr != "."];
  twentyfiveminuteslateFR = Cases[penday, {_, "FR", ___, x_, owg_, olh_,
    ofr_, cwg_, clh_, cfr_} /; x > cfr + 1200 && x <= cfr + 1500 && cfr != "."];
  thirtyminuteslateFR = Cases[penday, {_, "FR", ___, x_, owg_, olh_,
    ofr_, cwg_, clh_, cfr_} /; x > cfr + 1500 && x <= cfr + 1800 && cfr != "."];
  If[Length[fiveminuteslateFR] > 0, closingtimeFR = closingtimeFR + 300];
  If[Length[tenminuteslateFR] > 0, closingtimeFR = closingtimeFR + 300];
  If[Length[fifteenminuteslateFR] > 0, closingtimeFR = closingtimeFR + 300];
  If[Length[twentyminuteslateFR] > 0, closingtimeFR = closingtimeFR + 300];
  If[Length[twentyfiveminuteslateFR] > 0, closingtimeFR = closingtimeFR + 300];
  If[Length[thirtyminuteslateFR] > 0, closingtimeFR = closingtimeFR + 300];

  (*Here is a new part*)
  (*what if LH or FR open before WG? Can't be the case; WG time must be adapted*)
  If[openingtimeLH >= openingtimeFR, openingtimeLH = openingtimeFR];
  If[openingtimeWG >= openingtimeLH, openingtimeWG = openingtimeLH];
  If[closingtimeFR >= closingtimeLH, closingtimeLH = closingtimeFR];
  If[closingtimeLH >= closingtimeWG, closingtimeWG = closingtimeLH];

  (*End of new part*)

  If[Length[penday] > 0,

```

```

tpenday = Transpose[penday];
newpenday = Transpose[
  Join[tpenday[[1 ;; 9]],
    {Table[openingtimeWG, {Length[penday]}],
      Table[openingtimeLH, {Length[penday]}],
      Table[openingtimeFR, {Length[penday]}],
      Table[closingtimeWG, {Length[penday]}],
      Table[closingtimeLH, {Length[penday]}],
      Table[closingtimeFR, {Length[penday]}]}
  ]
],
newpenday = {};
];
timeshift = {openingtimeWG - penday[[1, 10]], openingtimeLH - penday[[1, 11]],
openingtimeFR - penday[[1, 12]], closingtimeWG - penday[[1, 13]],
closingtimeLH - penday[[1, 14]], closingtimeFR - penday[[1, 15]]};
AppendTo[alltimeshifts, timeshift];
AppendTo[openingtimesordered,
openingtimeWG <= openingtimeLH && openingtimeLH <= openingtimeFR &&
closingtimeFR <= closingtimeLH && closingtimeLH <= closingtimeWG];
newpenday = DeleteCases[newpenday, {_, "WG", ___, x_, owg_, olh_,
  ofr_, cwg_, clh_, cfr_} /; (NumberQ[owg] && x < owg)];
newpenday = DeleteCases[newpenday, {_, "LH", ___, x_, owg_, olh_,
  ofr_, cwg_, clh_, cfr_} /; (NumberQ[olh] && x < olh)];
newpenday = DeleteCases[newpenday, {_, "FR", ___, x_, owg_, olh_,
  ofr_, cwg_, clh_, cfr_} /; (NumberQ[ofr] && x < ofr)];
newpenday = DeleteCases[newpenday, {_, "WG", ___, x_, owg_, olh_,
  ofr_, cwg_, clh_, cfr_} /; (NumberQ[cwg] && x > cwg)];
newpenday = DeleteCases[newpenday, {_, "LH", ___, x_, owg_, olh_,
  ofr_, cwg_, clh_, cfr_} /; (NumberQ[clh] && x > clh)];
newpenday = DeleteCases[newpenday, {_, "FR", ___, x_, owg_, olh_,
  ofr_, cwg_, clh_, cfr_} /; (NumberQ[cfr] && x > cfr)];
newpenday = DeleteCases[newpenday, {_, "WG", ___, x_, ".",
  olh_, ofr_, cwg_, clh_, cfr_}];
newpenday = DeleteCases[newpenday, {_, "LH", ___, x_,
  owg_, ".", ofr_, cwg_, clh_, cfr_}];
newpenday = DeleteCases[newpenday, {_, "FR", ___, x_,
  owg_, olh_, ".", cwg_, clh_, cfr_}];
newpenday = DeleteCases[newpenday, {_, "WG", ___, x_,
  owg_, olh_, ofr_, ".", clh_, cfr_}];
newpenday = DeleteCases[newpenday, {_, "LH", ___, x_,
  owg_, olh_, ofr_, cwg_, ".", cfr_}];
newpenday = DeleteCases[newpenday, {_, "FR", ___, x_,
  owg_, olh_, ofr_, cwg_, clh_, "."}];
AppendTo[timeorder,

If[NumberQ[openingtimeWG] && NumberQ[openingtimeLH], openingtimeWG <= openingtimeLH,
True] && If[NumberQ[openingtimeLH] && NumberQ[openingtimeFR], openingtimeLH <=
openingtimeFR, True] && If[NumberQ[closingtimeWG] && NumberQ[closingtimeLH],
closingtimeWG >= closingtimeLH, True] && If[NumberQ[closingtimeLH] &&
NumberQ[closingtimeFR], closingtimeLH >= closingtimeFR, True]
];
AppendTo[alldelations, Length[penday] - Length[newpenday]];

```

```

      AppendTo[newdata, newpenday];
    ];
  ]
newdata = Flatten[newdata, 1];

Length[newdata]
1 219 658

Save["newdata", newdata]

allhens // Length
421

Union[newdata[[All, 2]]] // Length
411

```

These two figures show that 10 hens were never recorded leaving the barn. Though Sabine checked the RFID tags at the end and they were fine.

datelist

```

{Friday 02/09/2016, Friday 07/10/2016, Friday 09/09/2016, Friday 12/08/2016,
Friday 22/07/2016, Friday 23/09/2016, Friday 26/08/2016, Friday 29/07/2016,
Friday 30/09/2016, Monday 01/08/2016, Monday 03/10/2016, Monday 08/08/2016,
Monday 10/10/2016, Monday 11/07/2016, Monday 12/09/2016, Monday 15/08/2016,
Monday 18/07/2016, Monday 22/08/2016, Monday 25/07/2016, Monday 26/09/2016,
Saturday 01/10/2016, Saturday 03/09/2016, Saturday 08/10/2016, Saturday 10/09/2016,
Saturday 13/08/2016, Saturday 16/07/2016, Saturday 23/07/2016, Saturday 24/09/2016,
Saturday 27/08/2016, Saturday 30/07/2016, Sunday 04/09/2016, Sunday 11/09/2016,
Sunday 14/08/2016, Sunday 16/10/2016, Sunday 17/07/2016, Sunday 24/07/2016,
Sunday 25/09/2016, Sunday 28/08/2016, Thursday 01/09/2016, Thursday 06/10/2016,
Thursday 08/09/2016, Thursday 11/08/2016, Thursday 13/10/2016, Thursday 18/08/2016,
Thursday 21/07/2016, Thursday 22/09/2016, Thursday 25/08/2016, Thursday 28/07/2016,
Thursday 29/09/2016, Tuesday 04/10/2016, Tuesday 06/09/2016, Tuesday 07/06/2016,
Tuesday 11/10/2016, Tuesday 13/09/2016, Tuesday 16/08/2016, Tuesday 19/07/2016,
Tuesday 23/08/2016, Tuesday 26/07/2016, Tuesday 27/09/2016, Tuesday 30/08/2016,
Wednesday 05/10/2016, Wednesday 07/09/2016, Wednesday 10/08/2016,
Wednesday 12/10/2016, Wednesday 14/09/2016, Wednesday 17/08/2016,
Wednesday 20/07/2016, Wednesday 21/09/2016, Wednesday 24/08/2016,
Wednesday 27/07/2016, Wednesday 28/09/2016, Wednesday 31/08/2016}

```

```
Length[datelist]
```

```
72
```

Saving and Exporting the file “transitiondata”

```
Save["transitiondata", newdata]
```

```
Export["transitiondata.csv", newdata]
```

```
transitiondata.csv
```

The file “transitiondata” is the main data file, where all transitions recorded during 72 days, where all areas were accessible, are listed. Each line is a single antenna reading (one transition of a

single hen from one area to another). The 15 columns are given below : "tag" : Hex - number of the RFID tag, "hen" : unique hen ID number, "ranger" : categorization based on an initial rough classification (not further used), "pen" : pen ID number (11 - 14), "date", "absolut time" (in sec), "from" : the area from which the hen is coming (IN : indoors (barn), WG : wintergarden, LH : yard ("Laufhof"), FR : free - range), "to" : the area where the hen is going to (same as before), "time" : the time of the day (in seconds counted from 0 : 00 : 00), "WG open" : the time (in seconds from 0 : 00 : 00) when the WG was opened on that day for that pen, "LH open" : the time (in seconds from 0 : 00 : 00) when the LH was opened on that day for that pen, "FR open" : the time (in seconds from 0 : 00 : 00) when the FR was opened on that day for that pen, "WG closed" : the time (in seconds from 0 : 00 : 00) when the WG was closed on that day for that pen, "LH closed" : the time (in seconds from 0 : 00 : 00) when the LH was closed on that day for that pen, "FR closed" : the time (in seconds from 0 : 00 : 00) when the FR was closed on that day for that pen

Time Warping

Here I create for every hen a file of the following kind: each column represents one day, each line is one timepoint in intervals of 10 seconds. For all days and all hens the starttime is 7:25 and the endtime is 17:00. Cell entries are Integers 1-4 wit 1: indoors (barn), 2: wintergarden, 3: yard (Laufhof), 4: free-range. The first entry gives a decimal number, where the Integer part gives the hen ID number and three decimal places give the day number (from .001 for day one to .072 for day seventytwo).

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

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

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

```
datelist = datelist[[Ordering[Table[AbsoluteTime[{datelist[[i],  
{"DayName", " ", "Day", "/", "Month", "/", "Year"}]], {i, Length[datelist]}]]]]];
```

```
hens = Round[allhens[[All, 1]]];
```

```
SetDirectory["F://Chicken Research/ChickenRanging4/henlines4"]
```

```
F:\Chicken Research\ChickenRanging4\henlines4
```

```
For[p = 1, p ≤ Length[hens], p++,  
  hentransitiondata = Select[transitiondata, #[[2]] == hens[[p]] &];  
  henlines = {};  
  For[j = 1, j ≤ Length[datelist], j++,  
    henday = Select[hentransitiondata, #[[5]] == datelist[[j]] &][[All, 7 ;; 9]];  
    PrependTo[henday, {".", "IN", 26700}];  
    AppendTo[henday, {"IN", ".", 61201}];  
    line = {};  
    For[t = 1, t < Length[henday], t++,  
      start = henday[[t]];  
      stop = henday[[t + 1]];
```

```

secs = Round[(stop[[3]] - start[[3]])];
tab = Which[
  start[[2]] == "IN" && stop[[1]] == "IN", Table["IN", {Ceiling[secs]}],
  start[[2]] == "IN" && stop[[1]] == "WG",
Join[Table["IN", {Ceiling[secs/2]}], Table["WG", {Ceiling[secs/2]}]],
  start[[2]] == "IN" && stop[[1]] == "LH", Join[Table["IN", {Ceiling[secs/3]}],
Table["WG", {Ceiling[secs/3]}], Table["LH", {Ceiling[secs/3]}]],
  start[[2]] == "IN" && stop[[1]] == "FR", Join[Table["IN", {Ceiling[secs/4]}],
Table["WG", {Ceiling[secs/4]}], Table["LH", {Ceiling[secs/4]}],
Table["FR", {Ceiling[secs/4]}]],
  start[[2]] == "WG" && stop[[1]] == "IN", Join[Table["WG", {Ceiling[secs/2]}],
Table["IN", {Ceiling[secs/2]}]],
  start[[2]] == "WG" && stop[[1]] == "WG", Table["WG", {Ceiling[secs]}],
  start[[2]] == "WG" && stop[[1]] == "LH",
Join[Table["WG", {Ceiling[secs/2]}], Table["LH", {Ceiling[secs/2]}]],
  start[[2]] == "WG" && stop[[1]] == "FR", Join[Table["WG", {Ceiling[secs/3]}],
Table["LH", {Ceiling[secs/3]}], Table["FR", {Ceiling[secs/3]}]],
  start[[2]] == "LH" && stop[[1]] == "IN", Join[Table["LH", {Ceiling[secs/2]}],
Table["IN", {Ceiling[secs/2]}]],
  start[[2]] == "LH" && stop[[1]] == "WG", Join[Table["LH", {Ceiling[secs/2]}],
Table["WG", {Ceiling[secs/2]}]],
  start[[2]] == "LH" && stop[[1]] == "LH", Table["LH", {Ceiling[secs]}],
  start[[2]] == "LH" && stop[[1]] == "FR", Join[Table["LH", {Ceiling[secs/3]}],
Table["WG", {Ceiling[secs/3]}], Table["FR", {Ceiling[secs/3]}]],
  start[[2]] == "FR" && stop[[1]] == "IN", Join[Table["FR", {Ceiling[secs/4]}],
Table["LH", {Ceiling[secs/4]}], Table["WG", {Ceiling[secs/4]}],
Table["IN", {Ceiling[secs/4]}]],
  start[[2]] == "FR" && stop[[1]] == "WG", Join[Table["FR", {Ceiling[secs/3]}],
Table["LH", {Ceiling[secs/3]}], Table["WG", {Ceiling[secs/3]}]],
  start[[2]] == "FR" && stop[[1]] == "LH", Join[Table["FR", {Ceiling[secs/2]}],
Table["LH", {Ceiling[secs/2]}]],
  start[[2]] == "FR" && stop[[1]] == "FR", Table["FR", {Round[secs]}]
];
If[Length[tab] > 0 && Length[tab] - secs > 0,
  tab = Delete[tab,
  Partition[RandomSample[Range[1, Length[tab]], Length[tab] - secs], 1]]
];
AppendTo[line, tab]
]; (*end For t*)
line = Flatten[line];
line = Table[line[[i]], {i, 1, Length[line], 10}];
(*a point every 10 seconds*)
AppendTo[henlines, Prepend[line, hentransitiondata[[1, 2]] + j/1000.]]
]; (*end For j*)
henlines = henlines //. {"IN" -> 1, "WG" -> 2, "LH" -> 3, "FR" -> 4};
henlines = Transpose[henlines];
Save[StringJoin["newhenlines", ToString[Round[hens[[p]]]], henlines];
Export[StringJoin["newhenlines", ToString[Round[hens[[p]]]], ".csv"], henlines]
]; (*end For p*)

```

Graphic

Here are just two graphical examples for single hen-days:

```
expand[{y1_, y2_, x_}] := {{x, y1}, {x, y2}}
```

```
hd = Cases[transitiondata, {____, hens[[31], ____], datelist[[53], ____]}];
line = ReplaceRepeated[hd, {"IN" → 1, "WG" → 2, "LH" → 3, "FR" → 4}][[All, 7 ;; 9]];
Line /@ Table[
  {{line[[i, 3]], line[[i, 2]]}, {line[[i + 1, 3]], line[[i, 2]]}}, {i, 1, Length[line] - 1}];
Join[Line /@ expand /@ line, Line /@ Table[{{line[[i, 2]], line[[i, 3]]},
  {line[[i, 2]], line[[i + 1, 3]]}}, {i, 1, Length[line] - 1}]];
ranginggraph = Graphics[Join[{Red, Thick}, Line /@ expand /@ line,
  Line /@ Table[{{line[[i, 3]], line[[i, 2]]}, {line[[i + 1, 3]], line[[i, 2]]}},
    {i, 1, Length[line] - 1}]], AspectRatio → 0.1]
```



```
hd = Cases[transitiondata, {____, hens[[32], ____], datelist[[63], ____]}];
line = ReplaceRepeated[hd, {"IN" → 1, "WG" → 2, "LH" → 3, "FR" → 4}][[All, 7 ;; 9]];
Line /@ Table[
  {{line[[i, 3]], line[[i, 2]]}, {line[[i + 1, 3]], line[[i, 2]]}}, {i, 1, Length[line] - 1}];
Join[Line /@ expand /@ line, Line /@ Table[{{line[[i, 2]], line[[i, 3]]},
  {line[[i, 2]], line[[i + 1, 3]]}}, {i, 1, Length[line] - 1}]];
ranginggraph = Graphics[Join[{Red, Thick}, Line /@ expand /@ line,
  Line /@ Table[{{line[[i, 3]], line[[i, 2]]}, {line[[i + 1, 3]], line[[i, 2]]}},
    {i, 1, Length[line] - 1}]], AspectRatio → 0.1]
```



Fixing format in *newhenlines*

Here I am fixing the headline in a few files of *newhenlines*. (In some cases the first entry was interpreted as string, not as number).

```
SetDirectory["F://Chicken Research/ChickenRanging4/henlines4"]
F:\Chicken Research\ChickenRanging4\henlines4

For[i = 1, i ≤ Length[allhens], i++;
  hennumber = allhens[[i, 1]];
  nhl = Get[StringJoin["newhenlines", ToString[hennumber]]];
  If[NumberQ[Total[nhl[[1]]]], ,
    nhl[[1]] = Table[day + hennumber, {day, 0.001, 0.072, 0.001}];
    Save[StringJoin["newhenlines", ToString[hennumber]], nhl];
    Export[StringJoin["newhenlines", ToString[hennumber], ".csv"], nhl];
    Print[StringJoin["corrected hen number ", ToString[hennumber]]]
  ]
]
```

corrected hen number 22
corrected hen number 34
corrected hen number 39
corrected hen number 84
corrected hen number 98
corrected hen number 531
corrected hen number 721
corrected hen number 735
corrected hen number 851
corrected hen number 919