

Saving Excel Sheets from R

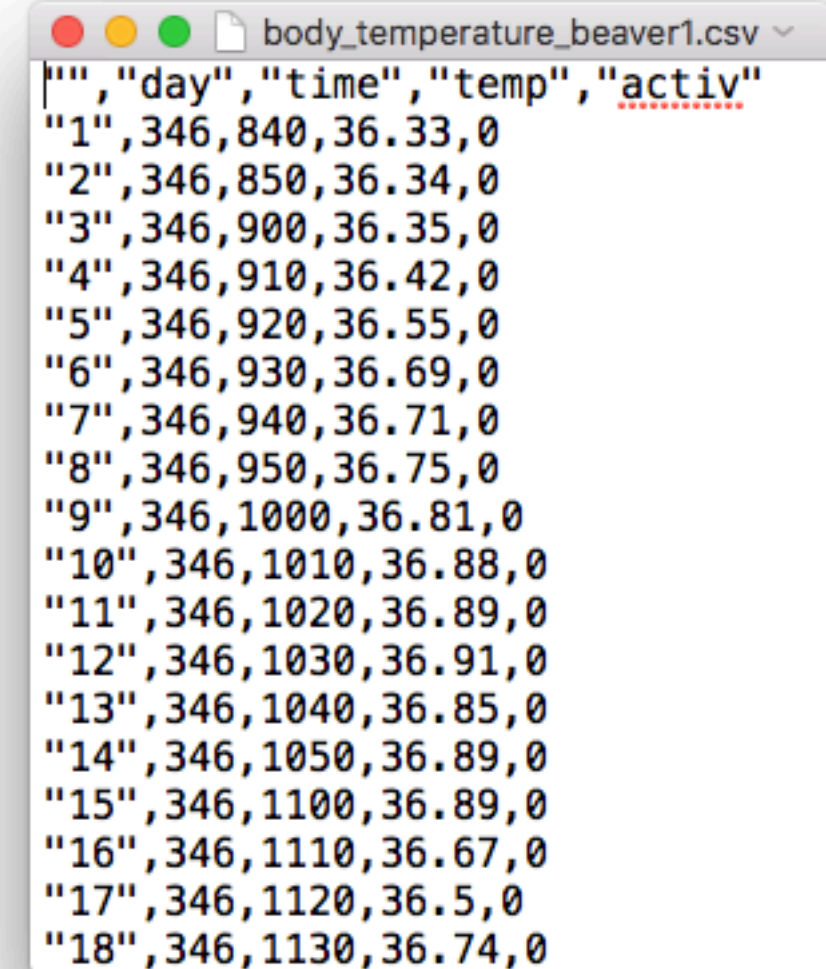
Erle Holgersen

November 28th, 2017

Basic Approach

- Write CSV file from R, import into Excel

```
write.csv(  
  beaver1,  
  'body_temperature_beaver1.csv'  
);
```

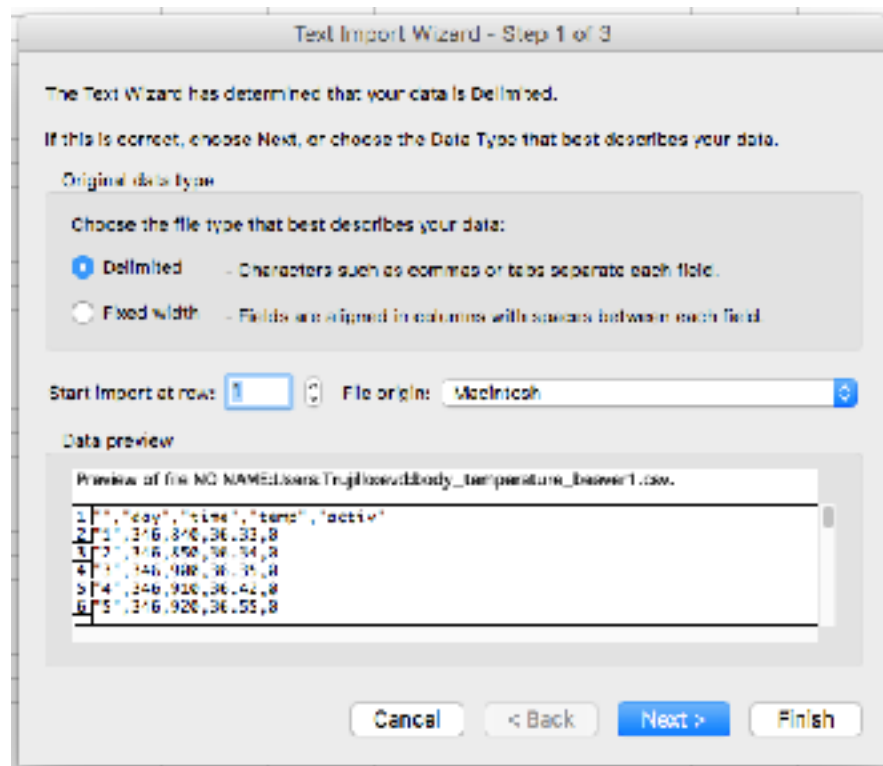


body_temperature_beaver1.csv

	day	time	temp	activ
"1"	346,840	36.33	0	
"2"	346,850	36.34	0	
"3"	346,900	36.35	0	
"4"	346,910	36.42	0	
"5"	346,920	36.55	0	
"6"	346,930	36.69	0	
"7"	346,940	36.71	0	
"8"	346,950	36.75	0	
"9"	346,1000	36.81	0	
"10"	346,1010	36.88	0	
"11"	346,1020	36.89	0	
"12"	346,1030	36.91	0	
"13"	346,1040	36.85	0	
"14"	346,1050	36.89	0	
"15"	346,1100	36.89	0	
"16"	346,1110	36.67	0	
"17"	346,1120	36.5	0	
"18"	346,1130	36.74	0	

Why Try Something Else?

- Data typically requires formatting after importing it
- If you know the end user will be using Excel, you can automate the process



	A	B	C	D	E
1		day	time	temp	activ
2	1	346	840	36.33	0
3	2	346	850	36.34	0
4	3	346	900	36.35	0
5	4	346	910	36.42	0
6	5	346	920	36.55	0
7	6	346	930	36.69	0
8	7	346	940	36.71	0
9	8	346	950	36.75	0
10	9	346	1000	36.81	0
11	10	346	1010	36.88	0
12	11	346	1020	36.89	0
13	12	346	1030	36.91	0
14	13	346	1040	36.85	0

Saving Excel Files

- Can use openxlsx package (or alternatives)
- Allows for saving directly to formatted xlsx sheets

```
library(openxlsx);
```

```
write.xlsx(  
  beaver1,  
  file = 'body_temperature_beaver1.xlsx'  
);
```

Making a Fancier Workbook

```
wb <- createWorkbook();
addWorksheet(wb, sheetName = 'Beaver 1');

writeData(
  wb, sheet = 'Beaver 1',
  x = t(c('Day', 'Time', 'Temperature (°C)', 'Active')),
  colNames = FALSE
);

writeData(
  wb, sheet = 'Beaver 1',
  x = beaver1, colNames = FALSE,
  startRow = 2);

# bold headers
addStyle(wb, sheet = 'Beaver 1', rows = 1, cols = 1:4,
  style = createStyle(textDecoration = 'bold')
);

saveWorkbook(wb, 'beavers.xlsx');
```

	A	B	C	D
1	Day	Time	Temperature (°C)	Active
2	346	840	36,33	0
3	346	850	36,34	0
4	346	900	36,35	0
5	346	910	36,42	0
6	346	920	36,55	0
7	346	930	36,69	0
8	346	940	36,71	0
9	346	950	36,75	0
10	346	1000	36,81	0
11	346	1010	36,88	0
12	346	1020	36,89	0
13	346	1030	36,91	0
14	346	1040	36,85	0
15	346	1050	36,89	0
16	346	1100	36,89	0
17	346	1110	36,67	0
18	346	1120	36,5	0
19	346	1130	36,74	0

Conclusion

- Non-programmers love Excel sheets
- Lots of packages available for saving directly to Excel sheets
- Depending on your use-case, it could be worth investing time in formatting output