Prepping datasets for CRF models

Running CRFs with categorical covariates requires expansion to model matrix format

The mosquito occurrence data from Golding et al 2015 (published in *Parasites & Vectors*) available from figshare here is useful for exploring how datasets need to be prepped for running Conditional Random Fields (CRF) models. Here, we will download the raw data from figshare (note, an internet connection will be needed for this step) change 'dipping_round' to a factor variable and remove un-needed columns

```
temp <- tempfile()
download.file('https://ndownloader.figshare.com/files/2075362',
              temp)
dataset <- read.csv(temp, as.is = T)
unlink(temp)
```

We can now change the categorical `dipping_round` and `field_site` variables to factors and remove some un-needed variables

```
dataset$dipping_round <- as.factor(dataset$dipping_round)
dataset$field_site <- as.factor(dataset$field_site)
dataset[,c(1,2,5,6)] <- NULL
```

It is important here to examine the level names of factor variables, as the 1st level (i.e. the dummy level) will be dropped from the dataset during conversion to model matrix format (as in standard `lme4` analysis of factor covariates)

```
levels(dataset$dipping_round)[1]
levels(dataset$field_site)[1]
```

The next step is to convert any factor variables into model matrix format. As mentioned above, this step will drop the first level of a factor and then create an additional column for each additional level (i.e. `dipping_round` levels `"3", "5" and "6"` will all be assigned their own unique columns, while `dipping_round` level `"2"` will be dropped and treated as the reference level). It is also convenient to change names of the new covariate columns so they are easier to view and interpret (done here using `dplyr::rename_all`)

```
library(dplyr)
analysis.data = dataset %>%
  cbind(.,data.frame(model.matrix(~.[,'field_site'],
                                  .)[,-1])) %>%
  cbind(.,data.frame(model.matrix(~.[,'dipping_round'],
                                  .)[,-1])) %>%
  dplyr::select(-field_site,-dipping_round) %>%
  dplyr::rename_all(funs(gsub("\\.|model.matrix", "", .)))
```

Finally, we need to convert species abundances to binary presence-absence format (as we are only estimating co-occurrences, not co-abundances). It is also highly advisable to scale any continuous variables so they all have `mean = 0` and `sd = 1`

```
analysis.data[, 1:16] <- ifelse(analysis.data[, 1:16] > 0, 1, 0)
analysis.data[, 17:20] <- scale(analysis.data[, 17:20], center = T, scale = T)
```