HW3 solutions

Eric Scott

1/30/2020

```
library(tibble)
library(dplyr)
library(ggplot2)
library(lubridate)
```

```
bflies <- tribble(
  ~year, ~date, ~site, ~fenders, ~silvery,
  2013, "25-Apr-13", "A",  0,  6,
  2013, "25-Apr-13", "B",  0,  4,
  2013, "25-Apr-13", "C",  0, 12,
  2013, "25-Apr-13", "D",  0,  5,
  2013,  "3-May-13", "A",  2,  6,
  2013,  "3-May-13", "B",  2,  6,
  2013,  "3-May-13", "C", 13,  6,
  2013,  "3-May-13", "D",  2, 16,
  2013, "14-May-13", "A",  8,  3,
  2013, "14-May-13", "B",  3,  4,
  2013, "14-May-13", "C",  8,  7,
  2013, "14-May-13", "D",  4,  3,
  2013, "20-May-13", "A",  6,  1,
  2013, "20-May-13", "B",  2,  7,
  2013, "20-May-13", "C",  2,  2,
  2013, "20-May-13", "D",  2,  1,
  2013, "31-May-13", "A",  5,  0,
  2013, "31-May-13", "B",  2,  0,
  2013, "31-May-13", "C", 10,  1,
  2013, "31-May-13", "D",  2,  2
)

head(bflies)
```

```
## # A tibble: 6 x 5
##    year date      site  fenders silvery
##   <dbl> <chr>     <chr>   <dbl>   <dbl>
## 1  2013 25-Apr-13 A           0       6
## 2  2013 25-Apr-13 B           0       4
## 3  2013 25-Apr-13 C           0      12
## 4  2013 25-Apr-13 D           0       5
## 5  2013 3-May-13  A           2       6
## 6  2013 3-May-13  B           2       6
```

```
# write.csv(bflies, here("data", "butterflies.csv"))
```

1. MLE for proportion of Fender's blues

```
bflies2 <- bflies %>%
  mutate(trials = fenders + silvery)

bflies2 %>%
  summarize_at(vars(fenders, silvery, trials), sum)
```

```
## # A tibble: 1 x 3
##   fenders silvery trials
##     <dbl>   <dbl>  <dbl>
## 1      73      92    165
```

```
mle_all <- 73/165
```

Confirm with glm()

```
m0 <- glm(cbind(fenders, silvery) ~ 1, family = binomial(link = "identity"), data = bflies)
coef(m0)
```

```
## (Intercept)
##   0.4424242
```

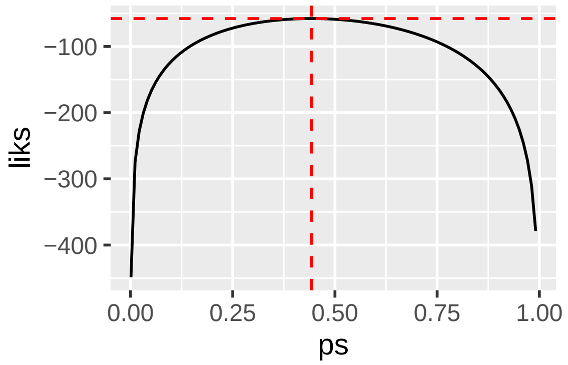
2. Log-likelihood profile

should have values of p on the x-axis and log-likelihood on y-axis

```
ps <- seq(0.001, 0.999, 0.01)
liks <- numeric(length = length(ps))
for(i in 1:length(ps)) {
   liks[i] <- dbinom(bflies2$fenders, bflies2$trials, ps[i], log = TRUE) %>% sum()
}

plotdata <- tibble(ps, liks)

#OR with mutate

plotdata2 <-
  tibble(ps) %>%
  rowwise %>%
  mutate(liks = dbinom(bflies2$fenders, bflies2$trials, ps, log = TRUE) %>% sum())

all(plotdata == plotdata2)
```

```
## [1] TRUE
```

```
ggplot(plotdata, aes(x = ps, y = liks)) +
  geom_line() +
  geom_hline(aes(yintercept = logLik(m0)), color = "red", lty = "dashed") +
  geom_vline(aes(xintercept = coef(m0)), color = "red", lty = "dashed")
```



3. Separate models for space and time

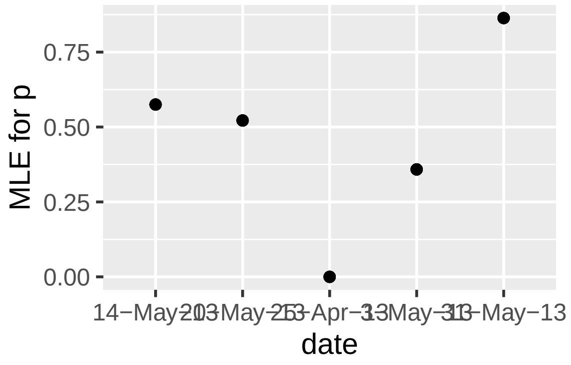
3a. proportion for each day

I did this with a mutate, but probably summarizing first would be better and more intuitive.

```
bflies3 <- bflies2 %>%
  group_by(date) %>%
  mutate(p_by_day = sum(fenders)/sum(trials)) %>%
  ungroup()

bflies3 %>% group_by(date) %>% summarize(first(p_by_day))
```

```
## # A tibble: 5 x 2
##   date      `first(p_by_day)`
##   <chr>                 <dbl>
## 1 14-May-13             0.575
## 2 20-May-13             0.522
## 3 25-Apr-13             0
## 4 3-May-13              0.358
## 5 31-May-13             0.864
```

```
ggplot(bflies3, aes(x =date, y = p_by_day)) +
  geom_point() + labs (y = "MLE for p") +
  labs(y = "MLE for p")
```



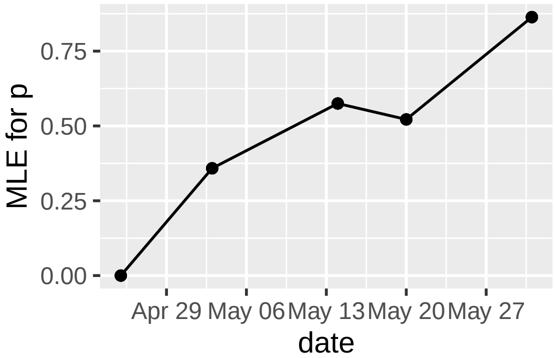
If you want lines, and you want the dates to be in order, probably the easiest solution is to use the `lubridate` package to convert `date` (a character vector) into an actual date vector, which is treated kind of like numeric.

```
class(bflies3$date)
```

```
## [1] "character"
```

```
bflies4 <-
  bflies3 %>%
  mutate(date = dmy(date)) #dmy is day, month, year.  There is a ymd() and a mdy() too.
```

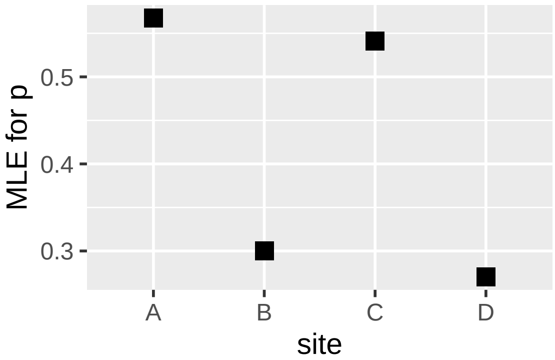
```
ggplot(bflies4, aes(x =date, y = p_by_day)) +
  geom_point() +
  geom_line() +
  labs(y = "MLE for p")
```



3b. By site

```
bflies5 <-
  bflies3 %>%
  group_by(site) %>%
  mutate(p_by_site = sum(fenders) / sum(trials)) %>%
  ungroup()
```

```
ggplot(bflies5, aes(x = site, y = p_by_site)) + geom_point(shape = "square", size = 3) +
  labs(y = "MLE for p")
```



4. log liks

```
bflies2 %>%
  #assuming different MLE p by date
  group_by(date) %>%
  mutate(p_by_date = sum(fenders) / sum(trials)) %>%
  #assuming different MLE p by site
  group_by(site) %>% #overrides previous group_by
  mutate(p_by_site = sum(fenders) / sum(trials)) %>%
  ungroup() %>%
  #assuming one MLE p for all the data
  add_column(p_all = mle_all) %>% #mle_all calculated earlier
  #calculate log-likelihoods with each model
  mutate(m1_logliks = dbinom(fenders, trials, p = mle_all, log = TRUE),
         m2_logliks = dbinom(fenders, trials, p_by_date, log = TRUE),
         m3_logliks = dbinom(fenders, trials, p_by_site, log = TRUE)) %>%
  #and add to get log-likelihood for entire dataset
  summarize(m1_logLik = sum(m1_logliks),
            m2_LogLik = sum(m2_logliks),
            m3_logLik = sum(m3_logliks))
```

```
## # A tibble: 1 x 3
##   m1_logLik m2_LogLik m3_logLik
##       <dbl>     <dbl>     <dbl>
## 1     -57.8     -31.0     -51.8
```

5. Nesting

Model b (dates) nested with model a (null). Model c (sites) nested with model a (null)

6. Which model is best?

The model for a separate proportion for each date has the highest log-likelihood. However, it also estimates 4 more parameters than the simplest model.

You could also do a LRT or compare AIC values.