Funky Predictions from lmer()

Andrew Zieffler

April 19, 2016

Read in Data and Load Libraries

```r
library(dplyr)
library(ggplot2)
library(lme4)
library(tidyr)

# Read in the data
nhl = read.csv(file = "/Users/andrewz/Documents/EPsy-8252/data/NHL-Wide.csv")

# Select variables
nhl2 = nhl %>%
  select(team, X2002, X2003, X2006, X2007, X2008, X2010, X2011, X2013, X2014, lat)

# Create long data from wide data (need tidyr package loaded)
nhlLong = nhl2 %>% gather(
  key = year, # Name of the variable that delineates the time points
  value = fci, # Name of the outcome
  c(X2002, X2003, X2006, X2007, X2008, X2010, X2011, X2013, X2014)
  )

# Remove the X from the years
nhlLong$year = sub("X", "", nhlLong$year)

# Coerce into a numeric value
nhlLong$year = as.numeric(nhlLong$year)

# Cebter the year predictor
nhlLong$c.year = nhlLong$year - mean(nhlLong$year)

# Examine the new data
str(nhlLong)
```

```
## 'data.frame':    279 obs. of  5 variables:
##  $ team  : Factor w/ 31 levels "Anaheim Ducks",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ lat   : num  33.8 33.5 33.8 42.4 42.9 ...
##  $ year  : num  2002 2002 2002 2002 2002 ...
##  $ fci   : num  212 214 254 293 223 ...
##  $ c.year: num  -6.22 -6.22 -6.22 -6.22 -6.22 ...
```

```r
head(nhlLong)
```

```
##                  team      lat year    fci    c.year
## 1     Anaheim Ducks 33.80815 2002 212.28 -6.222222
## 2   Arizona Coyotes 33.53193 2002 214.28 -6.222222
## 3 Atlanta Thrashers 33.75753 2002 253.51 -6.222222
## 4     Boston Bruins 42.36644 2002 293.49 -6.222222
## 5    Buffalo Sabres 42.88380 2002 222.84 -6.222222
## 6    Calgary Flames 51.03743 2002 178.87 -6.222222
```

Fit lmer() Model

```
model.f = lmer(fci ~ 1 + c.year + I(c.year^2) + lat + lat:c.year + lat:I(c.year^2) + (1 + c.year + I(c.y
```

Set Up Plotting Data

```
# Create data set
wild = data.frame(
    c.year = seq(from = -6.2, to = 5.8, by = 0.1),
    lat = 44.94509,
    team = "Minnesota Wild"
    )

stars = data.frame(
    c.year = seq(from = -6.2, to = 5.8, by = 0.1),
    lat = 32.79069,
    team = "Dallas Stars"
    )

avalanche = data.frame(
    c.year = seq(from = -6.2, to = 5.8, by = 0.1),
    lat = 39.74857,
    team = "Colorado Avalanche"
    )

plotData = rbind(avalanche, stars, wild)

# Compute y-hat values
plotData$yhat = predict(model.f, newdata = plotData)

# Examine first two predicted values
plotData %>% filter(c.year < -6.1)
```

```
##   c.year      lat               team     yhat
## 1   -6.2 39.74857 Colorado Avalanche 240.4837
## 2   -6.1 39.74857 Colorado Avalanche 240.0386
## 3   -6.2 32.79069       Dallas Stars 240.7508
## 4   -6.1 32.79069       Dallas Stars 240.1869
## 5   -6.2 44.94509     Minnesota Wild 245.7944
## 6   -6.1 44.94509     Minnesota Wild 247.3134
```

Model Predictions by Computation

Here we compute each team's predicted FCI without using `predict()`. The `coef()` function produces the model coefficients for each team.

```
mw = coef(model.f)$team %>% filter(row.names(.) == "Minnesota Wild")
ds = coef(model.f)$team %>% filter(row.names(.) == "Dallas Stars")
ca = coef(model.f)$team %>% filter(row.names(.) == "Colorado Avalanche")

# Bind them together into a matrix
all = matrix(c(mw, ds, ca), byrow = TRUE, nrow = 3)
all
```

```
##      [,1]     [,2]      [,3]     [,4]     [,5]      [,6]
## [1,] 74.01936 -12.96054 2.951346 5.657055 0.5417028 -0.07254555
## [2,] 44.63861 -15.48293 3.022645 5.657055 0.5417028 -0.07254555
## [3,] 14.7064  -17.30652 3.588965 5.657055 0.5417028 -0.07254555
```

For the Wild, the equation is:

$$\text{FCI} = 74.01936 - 12.960537(\text{c.year}) + 2.9513462(\text{c.year}^2) + 5.657055(\text{lat}) + 0.5417028(\text{lat})(\text{c.year}) - 0.07254555(\text{lat})(\text{c.year}^2)$$

Substituting in the Wild's latitude of 44.94509, we can compute the predicted FCI for `c.year` of -6.2 and -6.1.

```
c.year = c(-6.2, -6.1)
74.01936 - 12.960537*c.year + 2.9513462*(c.year^2) + 5.657055*44.94509 +
  0.5417028*44.94509*c.year  - 0.07254555*44.94509*(c.year^2)
```

```
## [1] 245.7945 247.3134
```

Similarly, we can compute the predicted FCI for the Avalanche and the Stars.

```
# Avalanche
14.70640 - 17.306516*c.year + 3.5889653*(c.year^2) + 5.657055*39.74857 +
  0.5417028*39.74857*c.year - 0.07254555*39.74857*(c.year^2)
```

```
## [1] 240.4837 240.0387
```

```
# Stars
44.63861 - 15.482935*c.year + 3.0226454*(c.year^2) + 5.657055*32.79069 +
  0.5417028*32.79069*c.year - 0.07254555*32.79069*(c.year^2)
```

```
## [1] 240.7508 240.1869
```

These correspond to the values computed when we use the `predict()` function.

Take 2 with the predict() Function

This time we will arrange the teams in `plotData` in a different order.

```
# Put Wild first, then CO, then Stars
plotData2 = rbind(wild, avalanche, stars)

# Compute y-hat values
plotData2$yhat = predict(model.f, newdata = plotData2)

# Examine first two predicted values
plotData2 %>% filter(c.year < -6.1)
```

```
##   c.year      lat               team     yhat
## 1   -6.2 44.94509     Minnesota Wild 237.9366
## 2   -6.1 44.94509     Minnesota Wild 238.2367
## 3   -6.2 39.74857 Colorado Avalanche 237.3404
## 4   -6.1 39.74857 Colorado Avalanche 237.7742
## 5   -6.2 32.79069       Dallas Stars 251.7520
## 6   -6.1 32.79069       Dallas Stars 251.5280
```

Using the same data (just arranged differently), and predicting from the same model, we get completely different predicted values! What happened?

Here the `predict()` function used the Wild's data in Colorado's fitted equation.

```
14.70640 - 17.306516*c.year + 3.5889653*(c.year^2) + 5.657055*44.94509 +
  0.5417028*44.94509*c.year - 0.07254555*44.94509*(c.year^2)
```

```
## [1] 237.9366 238.2367
```

Colorado is the first team of the three alphabetically. In `plotData2`, the Wild's data is first (level 1 in the team factor). Somehow R is using the first level (the Wild) in the first team's (Colorado's) equation. Why? I do not know. Probably something with how the `predict()` function is programmed.

Solution

There are several solutions: One is to make sure the teams (cluster variable) are in alphabetical order, Colorado Avalanche, Dallas Stars, Minnesota Wild, like it was in the initial `plotData`.

The second solution is to predict for each team individually and then bind them into a single data frame after you do the prediction.

```
wild = data.frame(
    c.year = seq(from = -6.2, to = 5.8, by = 0.1),
    lat = 44.94509,
    team = "Minnesota Wild"
    )
wild$yhat = predict(model.f, newdata = wild)

stars = data.frame(
    c.year = seq(from = -6.2, to = 5.8, by = 0.1),
    lat = 32.79069,
    team = "Dallas Stars"
    )
```

```r
stars$yhat = predict(model.f, newdata = stars)

avalanche = data.frame(
    c.year = seq(from = -6.2, to = 5.8, by = 0.1),
    lat = 39.74857,
    team = "Colorado Avalanche"
    )
avalanche$yhat = predict(model.f, newdata = avalanche)

# Bind into a data frame
plotData = rbind(avalanche, wild, stars)
plotData2 = rbind(wild, avalanche, stars)

# Examine first two predicted values
plotData %>% filter(c.year < -6.1)
```

```
##   c.year      lat               team     yhat
## 1   -6.2 39.74857 Colorado Avalanche 240.4837
## 2   -6.1 39.74857 Colorado Avalanche 240.0386
## 3   -6.2 44.94509     Minnesota Wild 245.7944
## 4   -6.1 44.94509     Minnesota Wild 247.3134
## 5   -6.2 32.79069       Dallas Stars 240.7508
## 6   -6.1 32.79069       Dallas Stars 240.1869
```

```r
plotData2 %>% filter(c.year < -6.1)
```

```
##   c.year      lat               team     yhat
## 1   -6.2 44.94509     Minnesota Wild 245.7944
## 2   -6.1 44.94509     Minnesota Wild 247.3134
## 3   -6.2 39.74857 Colorado Avalanche 240.4837
## 4   -6.1 39.74857 Colorado Avalanche 240.0386
## 5   -6.2 32.79069       Dallas Stars 240.7508
## 6   -6.1 32.79069       Dallas Stars 240.1869
```

In any case, you should always double-check any computer program's computations before submitting a paper (or assignment).