Lesson 25 Boardsheet - Multilevel Modeling

Kevin Cummiskey

3/26/2020

Reference:

- "Multilevel Modeling of OBP trajectories" by Jim Albert. https://baseballwithr.wordpress.com/2019/11/25/multilevel-modeling-of-obp-trajectories/

Learning Objectives:

- Gain appreciation for how Bayesian statistics can help us combine prior knowledge with new observations to update our beliefs.
- Gain appreciation for how multilevel modeling can "pool" information to arrive a better estimates for individuals.

Review

Let's say your goal is to predict a player's career trajectory. What are common issues that arise if you fit a quadratic model using only that player's data?

POLL

What are some ways Bayesian, multilevel models improve upon the individual trajectories?

POLL

- Bayesian - allows us to combine prior knowledge with new data to update beliefs
- Multilevel - borrows information from other players when making individual predictions

Prior Knowledge "League Avg" ⟷ Data "Individual"

Players who debuted in 2001.

Once again, we will look at players who debuted in 2001 and had at least 1000 career at bats. These players include Albert Pujols and Ichiro Suzuki.

```
library(Lahman)
library(tidyverse)
library(lubridate)
library(ggrepel)

#Players with debut in year 2000
Master %>%
  filter(year(debut) == 2001) %>%
  pull(playerID) -> year2001.ids

#Players with at least 1000 atbats
Batting %>%
  filter(playerID %in% year2001.ids) %>%
  group_by(playerID) %>%
  summarize(AB = sum(AB)) %>%
  filter(AB > 1000) %>%
  pull(playerID) -> player.ids

player.ids = c(player.ids, "martijd02", "judgeaa01")

# get statistics by age and add names
source("Chapter8_functions.R")
player.ids %>%
  map_df(get_stats) %>%
  left_join(Master %>% select(nameLast, nameFirst, playerID))-> player.stats
```

Trajectories from Multilevel Models

Now, let's fit trajectories that pool information from all the players in the data set.

```
library(brms)
library(rstan)
player.stats %>%
  mutate(AgeD = Age - 30,
         Player = paste(nameFirst,nameLast, sep = " ")) -> player.stats

fit <- brm(OB | trials(PA) ~ AgeD + I(AgeD ^ 2) +
             (AgeD + I(AgeD ^ 2) | Player),
           data = player.stats,
           family = binomial("logit"))

Player_Fits <- coef(fit)$Player[, "Estimate", ] %>%
  as_tibble(rownames = "Player") %>%
  rename(b0.hat = Intercept,
         b1.hat = AgeD,
         b2.hat = IAgeDE2)

#
```

```r
# merge these estimates with our main dataset
player.stats <- inner_join(player.stats, Player_Fits, by = "Player")

# find estimates of OBP probs at each age
# note plogis is the logit function
player.stats %>%
  mutate(OBP.pred = plogis(b0.hat + b1.hat * AgeD + b2.hat * AgeD^2)) -> player.stats
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Users/kfcummiskey/Library/
## In file included from <built-in>:1:
## In file included from /Users/kfcummiskey/Library/R/3.6/library/StanHeaders/include/stan/math/prim/ma
## In file included from /Users/kfcummiskey/Library/R/3.6/library/RcppEigen/include/Eigen/Dense:1:
## In file included from /Users/kfcummiskey/Library/R/3.6/library/RcppEigen/include/Eigen/Core:88:
## /Users/kfcummiskey/Library/R/3.6/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:1: error
## namespace Eigen {
## ^
## /Users/kfcummiskey/Library/R/3.6/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:16: erro
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Users/kfcummiskey/Library/R/3.6/library/StanHeaders/include/stan/math/prim/ma
## In file included from /Users/kfcummiskey/Library/R/3.6/library/RcppEigen/include/Eigen/Dense:1:
## /Users/kfcummiskey/Library/R/3.6/library/RcppEigen/include/Eigen/Core:96:10: fatal error: 'complex' 
## #include <complex>
##          ^~~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
##
## SAMPLING FOR MODEL 'cec340bfd9aff6b986bb28bfaed3b2e1' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000374 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 3.74 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 41.7992 seconds (Warm-up)
## Chain 1:                16.1934 seconds (Sampling)
## Chain 1:                57.9926 seconds (Total)
```

```
## Chain 1:
##
## SAMPLING FOR MODEL 'cec340bfd9aff6b986bb28bfaed3b2e1' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.000348 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 3.48 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 50.4576 seconds (Warm-up)
## Chain 2:                18.1397 seconds (Sampling)
## Chain 2:                68.5973 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'cec340bfd9aff6b986bb28bfaed3b2e1' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.000227 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 2.27 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 44.9567 seconds (Warm-up)
## Chain 3:                21.2189 seconds (Sampling)
## Chain 3:                66.1756 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'cec340bfd9aff6b986bb28bfaed3b2e1' NOW (CHAIN 4).
## Chain 4:
```

Albert Pujols

Figure 1: Pujols

```
## Chain 4: Gradient evaluation took 0.00047 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 4.7 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 46.8277 seconds (Warm-up)
## Chain 4:                15.0092 seconds (Sampling)
## Chain 4:                61.8368 seconds (Total)
## Chain 4:
```

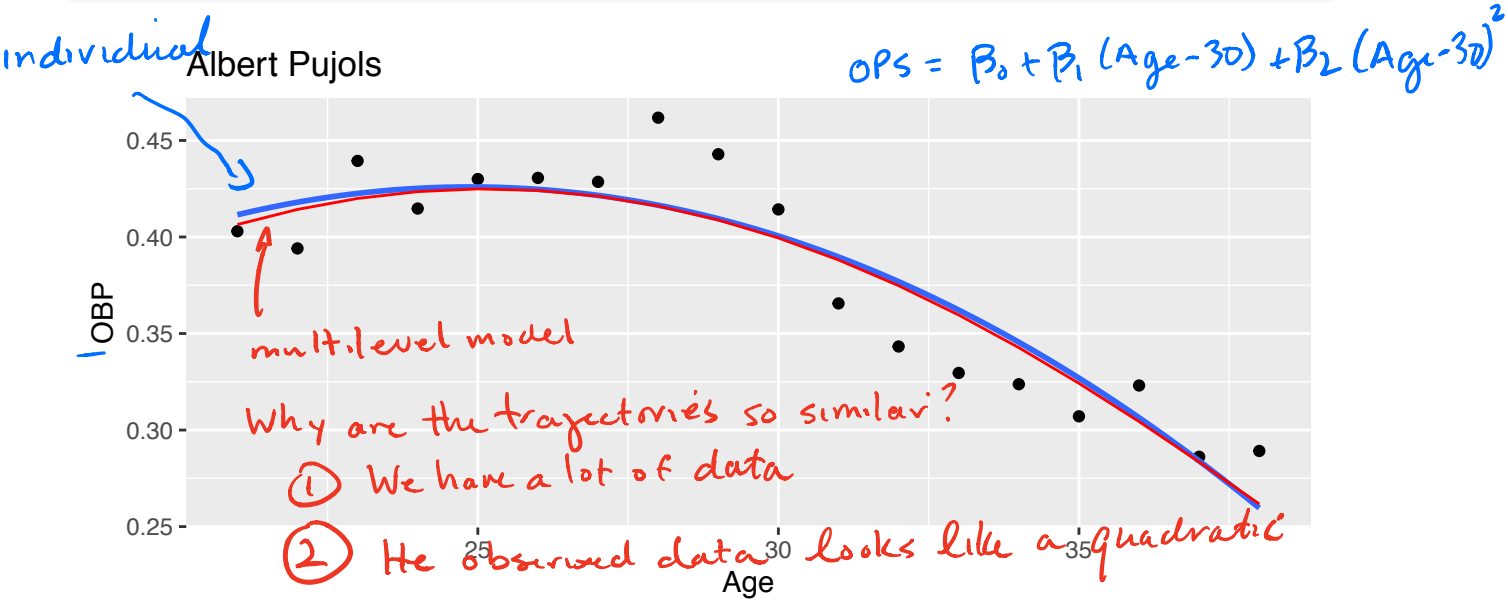
Albert Pujols

```
player.stats %>%
  filter(playerID == "pujolal01") %>%
  ggplot(aes(x = Age, y = OBP)) +
  geom_point() +
  geom_smooth(method = "lm",
```

Ichiro

Figure 2: Ichiro

```
              formula = y ~ x + I(x^2),
              se = FALSE) +
  geom_line(aes(y = OBP.pred), color = "red") +
  labs(title = "Albert Pujols")
```



Ichiro Suzuki

```
player.stats %>%
  filter(playerID == "suzukic01") %>%
  ggplot(aes(x = Age, y = OBP)) +
```

```
  geom_point() +
  geom_smooth(method = "lm",
              formula = y ~ x + I(x^2),
              se = FALSE) +
  geom_line(aes(y = OBP.pred), color = "red") +
  labs(title = "Ichiro Suzuki")
```

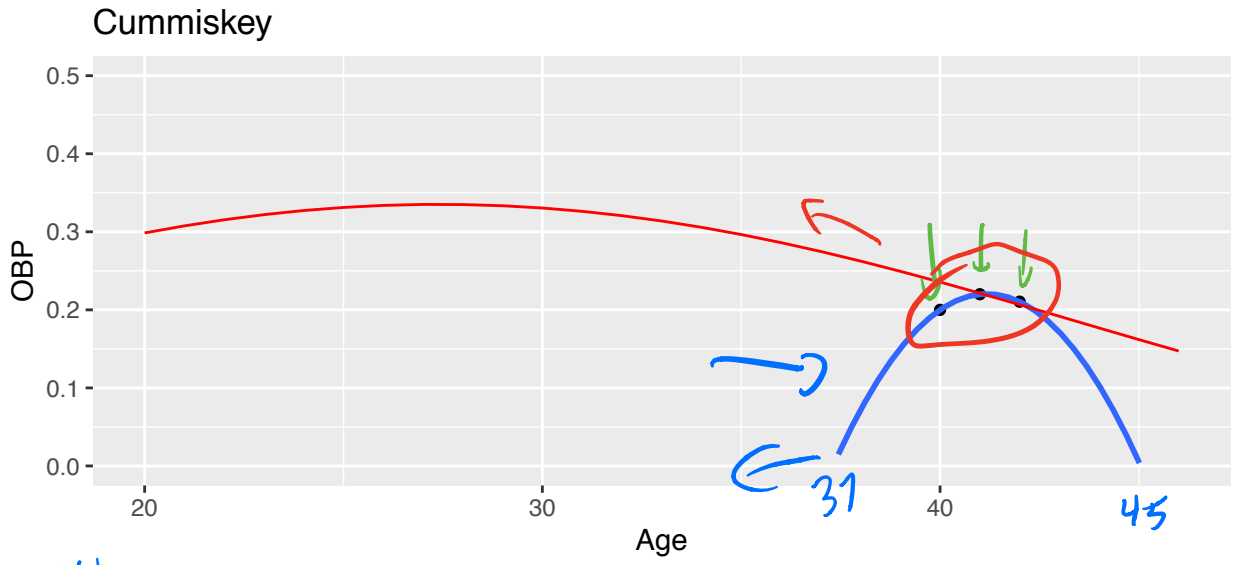
Figure 3: Cummiskey

Okay, let's have some fun.

Cummiskey

In my good fortune, it turns out a Red Sox scout was visiting West Point and took in a thrilling Army Math softball game. In a sheer stroke of luck, I made a great play at third base and the Red Sox signed me to a three year contract. I retire from the Army and play three seasons in my 40s. I put up some abysmal numbers – the fans boo me out of Boston ("you're a real chowda-head!!") and I retire at age 43.

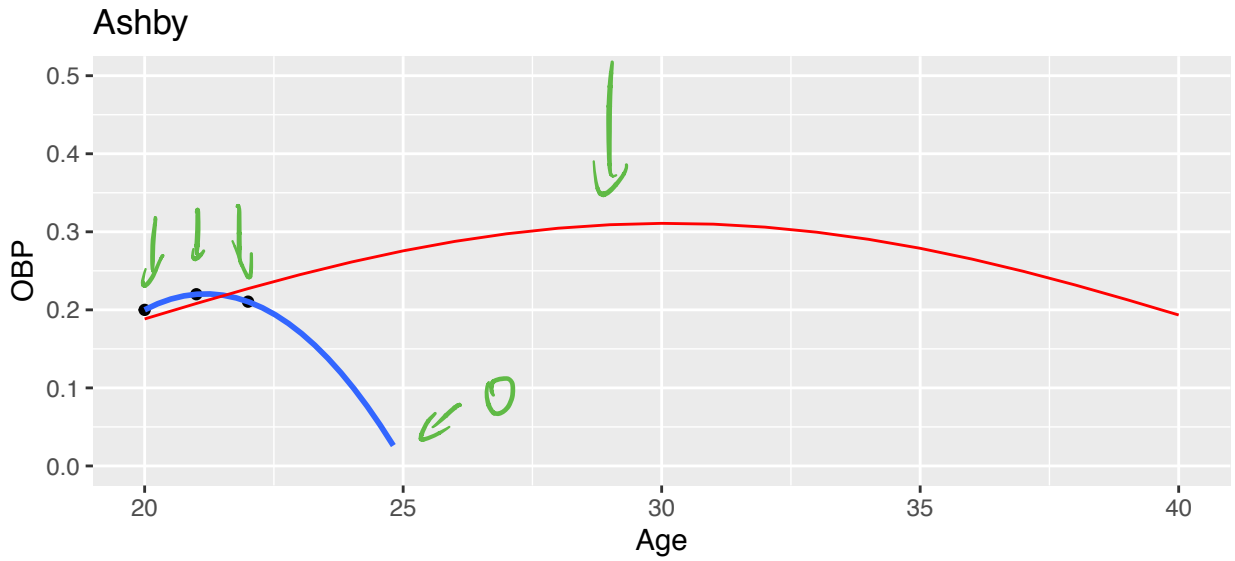
```
player.stats %>%
  filter(playerID == "cummiskey") %>%
  ggplot(aes(x = Age, y = OBP)) +
  geom_point() +
  geom_smooth(method = "lm",
              formula = y ~ x + I(x^2),
              se = FALSE,
              fullrange = TRUE) +
  geom_line(aes(y = OBP.pred), color = "red") +
  labs(title = "Cummiskey") + ylim(0,0.5)
```



Ashby

On the same scouting trip, the Red Sox spot a promising young outfielder named Nick Ashby. Nick leaves West Point and signs a contract with the Red Sox. He plays three seasons. In a really interesting twist of fate, he gets the exact same batting results as LTC Cummiskey, who Nick usually just refers to now as "that Chowda-head" at third base.

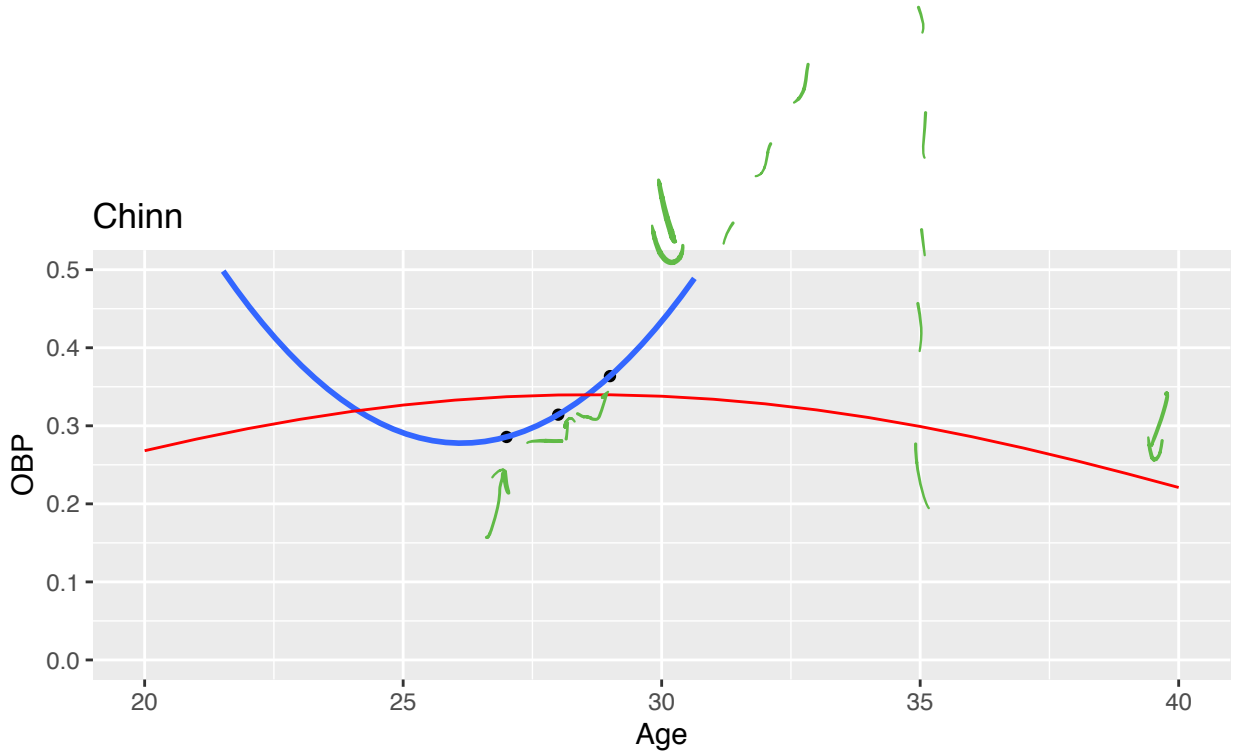
```
player.stats %>%
  filter(playerID == "ashby") %>%
  ggplot(aes(x = Age, y = OBP)) +
  geom_point() +
  geom_smooth(method = "lm",
              formula = y ~ x + I(x^2),
              se = FALSE,
              fullrange = TRUE) +
  geom_line(aes(y = OBP.pred), color = "red") +
  labs(title = "Ashby") + ylim(0,0.5)
```



Chinn

Samantha graduates from West Point and branches _____. Unknown to even her closest friends, she installs a batting cage in her basement and takes batting practice for 4 hours a day. After she serves her Army committment, she leaves the Army to pursue a career as in sports analytics working in the front office of the Boston Red Sox. One day, while the team is taking batting practice, a player chides her to hit some pitches (thinking some nerdy analyst has no chance hitting the ball). She hits five straight home runs and the Red Sox all of a sudden have a new DH.

```
player.stats %>%
  filter(playerID == "chinn") %>%
  ggplot(aes(x = Age, y = OBP)) +
  geom_point() +
  geom_smooth(method = "lm",
              formula = y ~ x + I(x^2),
              se = FALSE,
              fullrange = TRUE) +
  geom_line(aes(y = OBP.pred), color = "red") +
  labs(title = "Chinn") + ylim(0,0.5)
```

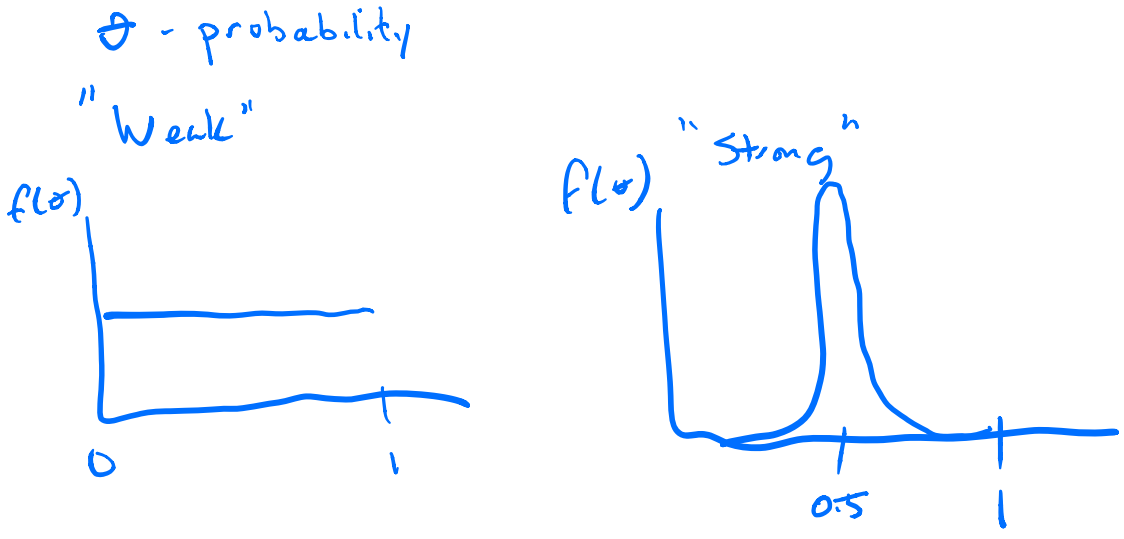


Efaw

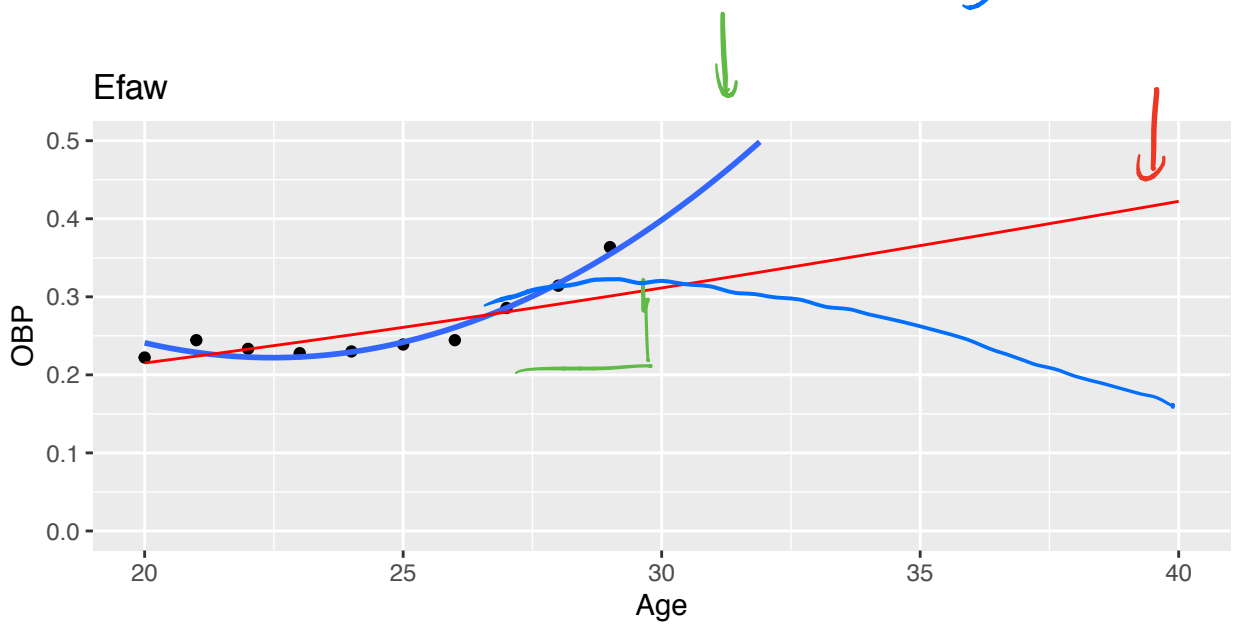
After being tortured for a semester by LTC Cummiskey in MA388, Andrew decides to spite him by signing a contract with the New York Yankees.

Ok, so these models do take some tuning.

```
player.stats %>%
  filter(playerID == "efaw") %>%
  ggplot(aes(x = Age, y = OBP)) +
  geom_point() +
  geom_smooth(method = "lm",
              formula = y ~ x + I(x^2),
              se = FALSE,
              fullrange = TRUE) +
  geom_line(aes(y = OBP.pred), color = "red") +
  labs(title = "Efaw") + ylim(0,0.5)
```



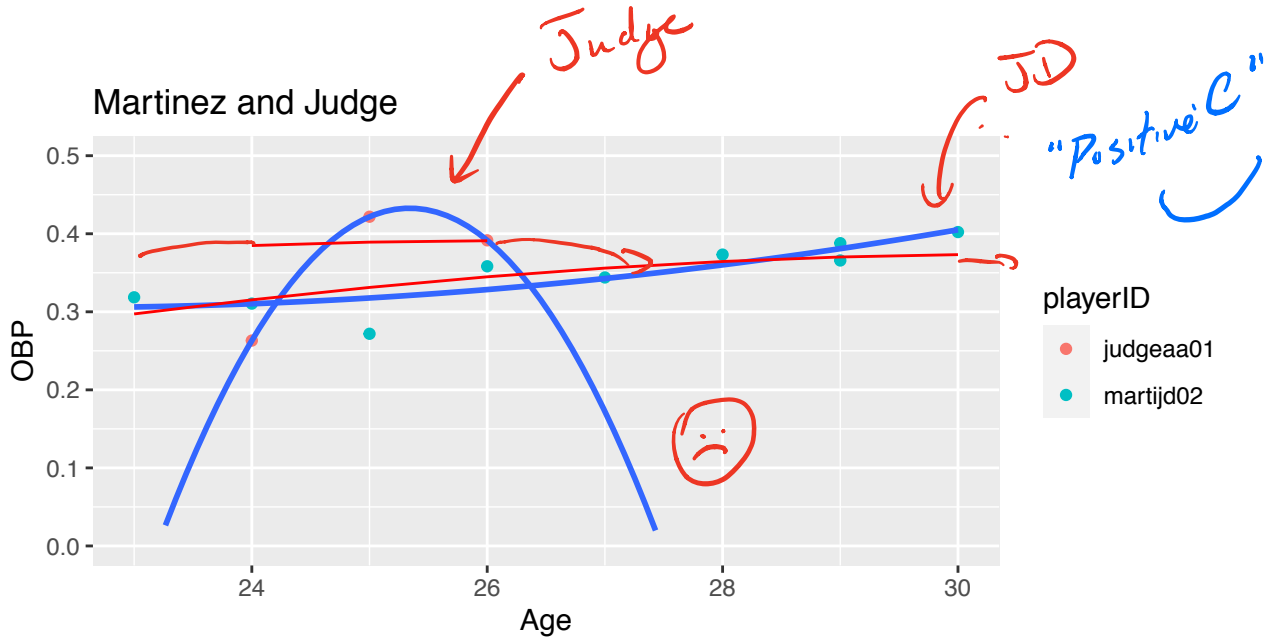
Tuning - how strong is our prior knowledge



Ok, let's be serious again.

Perhaps one of the more interesting duo's in MLB is J.D. Martinez (Red Sox - Age 32) and Aaron Judge (Yankees - Age 26)

```
player.stats %>%
  filter(playerID %in% c( "martijd02", "judgeaa01")) %>%
  ggplot(aes(x = Age, y = OBP, group = playerID)) +
  geom_point(aes(color = playerID)) +
  geom_smooth(method = "lm",
              formula = y ~ x + I(x^2),
              se = FALSE,
              fullrange = TRUE) +
  geom_line(aes(y = OBP.pred), color = "red") +
  labs(title = "Martinez and Judge") + ylim(0,0.5)
```



These models have the effect of "shrinking peak age" back towards the league average when there isn't much data on the player.

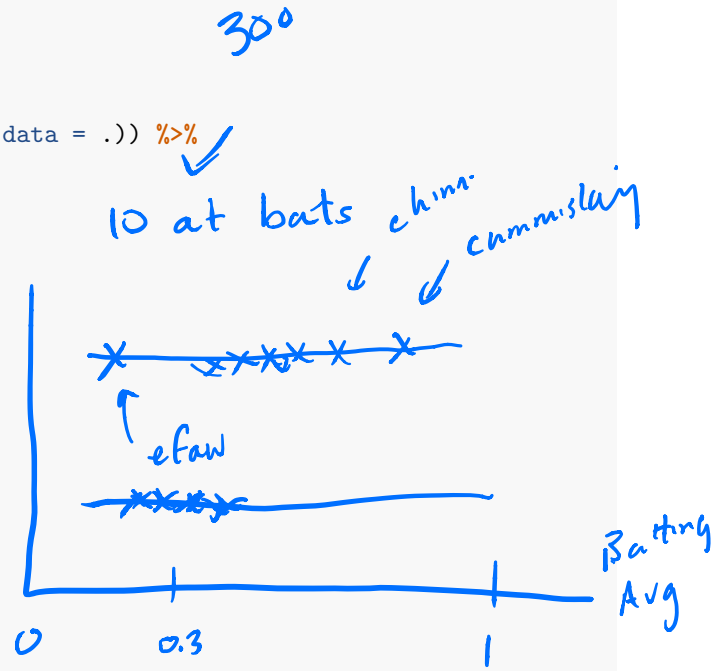
```
# find peak ages using two models
library(broom)
models <- player.stats %>%
  split(pull(.,playerID)) %>%
  map(~ lm(OBP ~ I(Age - 30) + I((Age-30)^2), data = .)) %>%
  map_df(tidy, .id = "playerID")

models %>%
  group_by(playerID) %>%
  summarize(A = estimate[1],
            B = estimate[2],
            C = estimate[3]) %>%
  mutate(Peak.age = 30 - B/2/C) %>%
  select(playerID, Peak.age) %>%
  mutate(Type = "individual") -> beta_coefs

player.stats %>%
  mutate(Peak.age = 30 - b1.hat/2/b2.hat,
         Type = "pooled") -> player.stats

beta_coefs %>%
  rbind(player.stats %>%
          select(playerID, Peak.age, Type) %>%
          distinct) -> peaks

ggplot(peaks, aes(x = Type, y = Peak.age)) +
  geom_point() +
  coord_flip() +
  ggtitle("Estimates of Peak Age") +
  ylim(20,40)
```



```
## Warning: Removed 4 rows containing missing values (geom_point).
```