HW 8 solutions

Your name here

Introduction

1. For batting average (`BA`), calculate the James-Stein estimate of the nine players career batting percentages using the `Batting.02` data as a base.

```
##    playerID     BA.JS
## 1 braunry02 0.3030739
## 2 caseyse01 0.2892500
## 3 ethiean01 0.2834621
## 4 greensh01 0.2871080
## 5 grudzma01 0.2925290
## 6  hallbi03 0.2687606
## 7 jonesja04 0.2837849
## 8 loducpa01 0.2774183
## 9 olivomi01 0.2599971
```

The James-Stein estimates are 0.303, ... 0.260 for the nine players, Ryan Braun through Miguel Olivo.

2. What is the relative amount of shrinkage towards the overall league average that we can expect for players' batting average?

Roughly 50% shrinkage towards the league average (50.4% to be exact, see code below), given the roughly 450 at bats.

3. Refer to the original Efron and Morris paper. Does this amount of shrinkage surprise you given their findings and their sample size?

Using 45 at bats, Efron and Morris found about 80% reversion towards the league average in a sample of roughly 15 players. Using 450 at bats, we would expect less reversion, which is indeed what we've found. However, with only 9 players, there may be a bit of noise around our estimated shrinkage of 50%.

4. Compare the root mean squared error of the J-S estimate and the MLE estimate (recall, the MLE is the players original batting average). Which is more accurate?

Slightly higher RMSE from the J-S estimate (0.01, compared to 0.009), so more accuracy from the MLE.

5. One of the players is Miguel Olivo (`playerID` = `olivomi01`), a catcher who batted 0.240 for his career. Catchers generally tend to hit worse than other position players. Explain how this may be impacting the accuracy of the J-S estimate.

Different positions in baseball no doubt hit at different averages. When linking all players together, our J-S estimator shrinks each player, no matter the position, to the league wide rate. In the case of Olivo, his number is `shrunk` towards a higher number representative of all positions. This is going to yield an overestimate of his true percentage.

6. Let's fix a players number of at bats. Given what we know about on base percentage and batting average, for which variable do you expect a larger reversion towards the overall league-wide average?

On-base is more repeatable - batting average will revert towards the league-wide average more than OBP.

7. *Extra credit*: Repeat the same analysis, except for OBP. Do your findings match your expectations from Question 6? For OBP, is the J-S estimate more accurate than the MLE estimate?

There's only about 23% reversion with OBP using the same sample and the same number of at bats. This matches our expectation that a players OBP is more representative of his true talent. As in batting average, J-S and MLE estimates are roughly equally accurate at predicting future performance.

Here's code for BA

```
k <- nrow(first.players)
k
```

```
## [1] 9
```

```
p.bar <- mean(first.players$BA)
p.bar
```

```
## [1] 0.2828204
```

```
p.hat <- first.players$BA
p.hat
```

```
## [1] 0.3237251 0.2958057 0.2841163 0.2914798 0.3024283 0.2544248 0.2847682
## [8] 0.2719101 0.2367257
```

```
ss.p.bar <- sum((p.hat - p.bar)^2)
ss.p.bar
```

```
## [1] 0.005356807
```

```
sigma.sq <- p.bar*(1-p.bar)/450 ##Rough approximation
sigma.sq
```

```
## [1] 0.0004507401
```

```
variance.ratio <- (k-3)*sigma.sq/ss.p.bar
variance.ratio
```

```
## [1] 0.5048605
```

```
c <-  1 - variance.ratio
c
```

```
## [1] 0.4951395
```

```
first.players$BA.MLE <- first.players$BA
first.players$BA.JS <- p.bar + c*(p.hat - p.bar)
first.players$BA.JS
```

```
## [1] 0.3030739 0.2892500 0.2834621 0.2871080 0.2925290 0.2687606 0.2837849
## [8] 0.2774183 0.2599971
```

```
all.players <- Batting.1 %>%
  group_by(playerID) %>%
  filter(playerID %in% first.players$playerID, yearID >= 2002) %>%
  summarise(BA.Career = sum(H)/sum(AB), OBP.Career = sum(H+BB+HBP)/sum(AB))
first.players1 <- inner_join(first.players, all.players) %>%
  select(playerID, BA, BA.MLE, BA.JS, BA.Career)
first.players1[,2:5] <- round(first.players1[,2:5], 3)
first.players1
```

```
##    playerID    BA BA.MLE BA.JS BA.Career
## 1 braunry02 0.324  0.324 0.303     0.306
## 2 caseyse01 0.296  0.296 0.289     0.297
## 3 ethiean01 0.284  0.284 0.283     0.285
## 4 greensh01 0.291  0.291 0.287     0.281
## 5 grudzma01 0.302  0.302 0.293     0.296
## 6  hallbi03 0.254  0.254 0.269     0.248
## 7 jonesja04 0.285  0.285 0.284     0.275
## 8 loducpa01 0.272  0.272 0.277     0.283
## 9 olivomi01 0.237  0.237 0.260     0.240
```

```
RMSE <- function(x, y){sqrt(mean((x-y)^2))}
RMSE(first.players1$BA.MLE, first.players1$BA.Career)
```

```
## [1] 0.008993825
```

```
RMSE(first.players1$BA.JS, first.players1$BA.Career)
```

```
## [1] 0.01095445
```

Here's code for OBP

```
k <- nrow(first.players)
k
```

```
## [1] 9
```

```
p.bar <- mean(first.players$OBP)
p.bar
```

```
## [1] 0.3630657
```

```
p.hat <- first.players$OBP
p.hat
```

```
## [1] 0.4035477 0.3863135 0.3959732 0.3856502 0.3708609 0.3495575 0.3642384
## [8] 0.3393258 0.2721239
```

```
ss.p.bar <- sum((p.hat - p.bar)^2)
ss.p.bar
```

```
## [1] 0.01285081
```

```
sigma.sq <- p.bar*(1-p.bar)/450 ##Rough approximation
sigma.sq
```

```
## [1] 0.0005138866
```

```
variance.ratio <- (k-3)*sigma.sq/ss.p.bar
variance.ratio
```

```
## [1] 0.2399319
```

```
c <-  1 - variance.ratio
c
```

```
## [1] 0.7600681
```

```
first.players$OBP.MLE <- first.players$OBP
first.players$OBP.JS <- p.bar + c*(p.hat - p.bar)


all.players <- Batting.1 %>%
  group_by(playerID) %>%
  filter(playerID %in% first.players$playerID, yearID >= 2002) %>%
  summarise(OBP.Career = sum(H)/sum(AB), OBP.Career = sum(H+BB+HBP)/sum(AB))
first.players1 <- inner_join(first.players, all.players) %>%
  select(playerID, OBP, OBP.MLE, OBP.JS, OBP.Career)
first.players1[,2:5] <- round(first.players1[,2:5], 3)
first.players1
```

```
##    playerID   OBP OBP.MLE OBP.JS OBP.Career
## 1 braunry02 0.404   0.404  0.394      0.407
## 2 caseyse01 0.386   0.386  0.381      0.395
## 3 ethiean01 0.396   0.396  0.388      0.405
## 4 greensh01 0.386   0.386  0.380      0.405
## 5 grudzma01 0.371   0.371  0.369      0.360
## 6  hallbi03 0.350   0.350  0.353      0.338
## 7 jonesja04 0.364   0.364  0.364      0.354
## 8 loducpa01 0.339   0.339  0.345      0.363
## 9 olivomi01 0.272   0.272  0.294      0.290
```

```
RMSE <- function(x, y){sqrt(mean((x-y)^2))}
RMSE(first.players1$OBP.MLE, first.players1$OBP.Career)
```

```
## [1] 0.01413035
```

```
RMSE(first.players1$OBP.JS, first.players1$OBP.Career)
```

```
## [1] 0.015
```