Lab 9

Michael Lopez, Skidmore College

Overview

In this lab, we'll gain more experience with player-level statistics in the NHL, while also gaining practical experience in implementing the James-Stein estimator. First, our data, and a review of what we covered in lecture.

```
library(RCurl)
library(mosaic)
url <- getURL("https://raw.githubusercontent.com/statsbylopez/StatsSports/master/NHL.csv")
nhl.data <- read.csv(text = url, stringsAsFactors = FALSE)
nhl.data <- filter(nhl.data, TOI > 500)
nhl.data <- na.omit(nhl.data)
nhl.data$ShP <- nhl.data$Goals/nhl.data$Shots
first <- filter(nhl.data, Season==20122013)
first.players <- first %>%
  group_by(Name) %>%
  filter(Shots <= 106, Shots >= 100, Position!="D") %>%
  select(Name, Position, Goals, Shots, ShP)
first.players
```

Next, some parameters for the J-S estimator.

```
k <- nrow(first.players)
k
p.bar <- mean(first.players$ShP)
p.bar
p.hat <- first.players$ShP
p.hat
ss.p.bar <- sum((p.hat - p.bar)^2)
ss.p.bar
sigma.sq <- p.bar*(1-p.bar)/103 ##Rough approximation
sigma.sq
```

1. In words, interpret `k`, `p.bar`, `p.hat`, `ss.p.bar`, `sigma.sq` in context of this example.

Building on our code from class, I introduce the ratio of variances comparing the within shooter variance and the between shooter variance. This variable is stored as `variance.ratio`, and represents the amount of shrinkage that we'll see towards the overall league average.

```
variance.ratio <- (k-3)*sigma.sq/ss.p.bar
variance.ratio
c <-  1 - variance.ratio
c
```

We use `c` to obtain the J-S estimate of each player's true shooting percentage.

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

As a final step, we link to each players career percentage. There's one nifty step of code in here: the `%in%` line, which returns TRUE or FALSE dependent on if a string lies in a larger vector of several strings.

As one example,

```
temp.list <- c("Steven", "Late", "Asleep")
x <- "Steven"
y <- "Paul"
x %in% temp.list
y %in% temp.list
```

In this case, because `Steven` is in `temp.list` and `Paul` is not, we see the returns of TRUE and FALSE.

Here's the next set of code.

```
all.players <- nhl.data %>%
  group_by(Name) %>%
  filter(Name %in% first.players$Name, Season>=20122013) %>%
  summarise(ShP.Career = sum(Goals)/sum(Shots), n.shots = sum(Shots))
first.players1 <- inner_join(first.players, all.players) %>%
  select(Name, ShP, ShP.MLE, ShP.JS, ShP.Career)
first.players1[,2:5] <- round(first.players1[,2:5], 3)
first.players1
```

2. What does the `%in%` command do in the above set of code?

One way of evaluating accuracy is the root mean squared error.

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

3. Change the function above to obtain the mean absolute error for each shooting percentage estimate. Also, how can these be interpreted? **Note**: `abs(-3)` returns 3.

Our final step is to visualize the J-S estimate.

```
a = matrix(rep(1:2, nrow(first.players1)), 2, nrow(first.players1))
b = matrix(c(first.players1$ShP, first.players1$ShP.JS),
           2, nrow(first.players1), byrow=TRUE)
matplot(a, b, pch=" ", ylab="", xaxt="n", xlim=c(0.5, 2.1),
        xlab = "", ylim=c(0, 0.2), main = "Shot Percentage")
matlines(a, b, lty = 2); matpoints(a, b, pch = 16)
text(rep(0.7, nrow(first.players1)), first.players1$ShP, first.players1$Name, cex=0.7)
text(1, 0.02, "First 100\nat shots", cex=0.7)
text(2, 0.02, "J-S\nestimator", cex=0.7)
```

And we can also compare both estimates to the player's career shooting percentage.

```r
a = matrix(rep(1:3, nrow(first.players1)), 3, nrow(first.players1))
b = matrix(c(first.players1$ShP, first.players1$ShP.JS, first.players1$ShP.Career),
           3, nrow(first.players1), byrow=TRUE)
matplot(a, b, pch=" ", ylab="", xaxt="n", xlim=c(0.2, 3.1),
        xlab = "", ylim=c(0, 0.2), main = "Shot Percentage")
matlines(a, b, lty = 2); matpoints(a, b, pch = 16)
text(rep(0.5, nrow(first.players1)), first.players1$ShP, first.players1$Name, cex=0.7)
text(1, 0.02, "First 100\nat shots", cex=0.7)
text(2, 0.02, "J-S\nestimator", cex=0.7)
text(3, 0.02, "Career\naverage", cex=0.7)
```

Broadening our scope

One next logical step is to estimate the relative amount of shrinkage towards the league-wide average that we would see among other players with different numbers of shots. Additionally, we can compare the RMSE at other cutoffs for both the original estimate ($p_{\hat{MLE}}$) and the James-Stein estimate.

4. Fill out the following table by plugging in the lower (`LB`) and upper (`UB`) bounds for number of shots in the 2012-2013 season. The code following should work.

LB	UB	Shrinkage	RMSE(MLE)	RMSE(JS)	n
42	48				
70	75				
90	95				
100	106	40%	3.7%	2.7%	12
110	120				
130	150				

```
LB <- 100 #Minimum number of shots
UB <- 106 #Minimum number of shots
first <- filter(nhl.data, Season==20122013)
first.players <- first %>%
  group_by(Name) %>%
  filter(Shots >= LB, Shots <= UB, Position!="D") %>%
  select(Name, Position, Goals, Shots, ShP)
first.players
k <- nrow(first.players)
k
p.bar <- mean(first.players$ShP)
p.bar
p.hat <- first.players$ShP
p.hat
ss.p.bar <- sum((p.hat - p.bar)^2)
ss.p.bar
sigma.sq <- p.bar*(1-p.bar)/(mean(LB, UB))
sigma.sq

variance.ratio <- (k-3)*sigma.sq/ss.p.bar
variance.ratio
c <-  1 - variance.ratio
c
first.players$ShP.MLE <- first.players$ShP
first.players$ShP.JS <- p.bar + c*(p.hat - p.bar)
all.players <- nhl.data %>%
  group_by(Name) %>%
  filter(Name %in% first.players$Name, Season>=20122013) %>%
  summarise(ShP.Career = sum(Goals)/sum(Shots), n.shots = sum(Shots))
first.players1 <- inner_join(first.players, all.players) %>%
  select(Name, ShP, ShP.MLE, ShP.JS, ShP.Career)
RMSE(first.players1$ShP.MLE, first.players1$ShP.Career)
RMSE(first.players1$ShP.JS, first.players1$ShP.Career)
```

Taking on a new variable.

All of the above was predicated on using a players shooting percentage (Goals/Shots) as the outcome. Of course, hockey officials may be interested in how much other variables shrink towards the overall league rate.

Our new outcome will be assists per 60 minutes, defined as follows:

```
nhl.data$A_Sixty <- nhl.data$Assists/nhl.data$TOI*60
```

Note that we already have this variable stored as a different name; we'll stick with `A_Sixty` as it goes beyond two decimal places.

5. We calculated shot percentage using bounds on the players total number of shots. What metric makes the most sense as far as bounding a players `A_Sixty`?

In the code below, we'll generate bounds using time-on-ice. This makes sense as its the denominator in the calcuation of `A_Sixty`.

First, a similar code to before:

```
LB <- 800 #Minimum number of shots
UB <- 850 #Minimum number of shots
first <- filter(nhl.data, Season==20122013)
first.players <- first %>%
  group_by(Name) %>%
  filter(TOI >= LB, TOI <= UB, Position!="D") %>%
  select(Name, Position, Goals, Shots, TOI, A_Sixty)
first.players
k <- nrow(first.players)
k
p.bar <- mean(first.players$A_Sixty)
p.bar
p.hat <- first.players$A_Sixty
p.hat
ss.p.bar <- sum((p.hat - p.bar)^2)
ss.p.bar
```

6. Describe `k`, `p.bar`, and `ss.p.bar` in the context of this metric. *Note*: perhaps `p.bar` may not be the most appropriate symbol!

Next, we'll calculate `c` and the variance ratio. To do so, we'll assume the the variance in an individuals assist rate is roughly 0.13 (there are more concrete ways to do this, which we may discuss later).

```
sigma.sq <- 0.13
sigma.sq

variance.ratio <- (k-3)*sigma.sq/ss.p.bar
variance.ratio
c <-  1 - variance.ratio
c


first.players$A_Sixty.MLE <- first.players$A_Sixty
first.players$A_Sixty.JS <- p.bar + c*(p.hat - p.bar)
all.players <- nhl.data %>%
  group_by(Name) %>%
  filter(Name %in% first.players$Name, Season>=20122013) %>%
  summarise(A_Sixty.Career = sum(Assists)/sum(TOI)*60)
first.players1 <- inner_join(first.players, all.players) %>%
  select(Name, A_Sixty, A_Sixty.MLE, A_Sixty.JS, A_Sixty.Career)
RMSE(first.players1$A_Sixty.MLE, first.players1$A_Sixty.Career)
RMSE(first.players1$A_Sixty.JS, first.players1$A_Sixty.Career)
```

7. How does the RMSE of each estimate compare?

And here's a plot:

```
a = matrix(rep(1:3, nrow(first.players1)), 3, nrow(first.players1))
b = matrix(c(first.players1$A_Sixty, first.players1$A_Sixty.JS, first.players1$A_Sixty.Career),
           3, nrow(first.players1), byrow=TRUE)
matplot(a, b, pch=" ", ylab="", xaxt="n", xlim=c(0.2, 3.1),
        xlab = "", ylim=c(0, 1.8), main = "Assist Rate")
matlines(a, b, lty = 2); matpoints(a, b, pch = 16)
text(rep(0.5, nrow(first.players1)), first.players1$A_Sixty, first.players1$Name, cex=0.7)
text(1, 0.02, "2012-13 season", cex=0.7)
text(2, 0.02, "J-S\nestimator", cex=0.7)
text(3, 0.02, "Career\naverage", cex=0.7)
```

8. In words, explain what is happening in the graph to assist rate (`A_Sixty`).
9. Identify individual-level metrics in other sports than may tend to revert towards the overall league-wide average over time.