1. (a) The MSE for the James-Stein estimator was 0.0213098 and the MSE for the 45 observations is 0.075374. Both are calculated relative to the overall season averages.

   (b) The MSE for the James-Stein estimator is smaller than that of the 45 observations which is what the paper suggested would happen.

2. (a) To generate my 18 baseball players, given that the typical batting averages are between 0.15 and 0.35, I will generate 18 random numbers from the UNIF(0.15,0.35) distribution.

   The realizations are 0.3413431, 0.192491, 0.2910382, 0.2263858, 0.2319875, 0.2550947, 0.1638004, 0.3434491, 0.2886289, 0.2088253, 0.2324988, 0.1632743, 0.2576372, 0.2775417, 0.2372611, 0.3248319, 0.3217733, 0.2858177.

   (b) To generate 45 at bats for each player, I will draw 45 times randomly from a BERN(y_i) distribution, which assumes each player is just as likely to get hits as not hits. The y_i's are the random draws from the Uniform distribution in part (a).

   The generated means from each player's 45 at bats are 0.4222222, 0.1333333, 0.1555556, 0.3111111, 0.2888889, 0.2222222, 0.2, 0.3777778, 0.3111111, 0.2222222, 0.2222222, 0.1333333, 0.3111111, 0.3111111, 0.3333333, 0.3111111, 0.2444444, 0.2444444.

   (c) The MSE for the Stein estimator from the 45 randomly generated observations per player was 0.0412951 while the MSE for the observed average of the data was 0.1302355. Stein does better in this case.

   (d) Repeat this entire procedure 1000 times and record the proportion of simulations where the James-Stein estimator is better.

   In 1000 trials, the Stein estimator had a lower MSE 99.9 % of the time.

   (e) *Summarize your results.*

   Stein's theorem is upheld in our simulation. For the baseball scenario, Stein's theorm says that the James-Stein estimator will predict the future averages more accurately in terms of MSE no matter what the true batting abilities are.

   We saw that this was true 99.9 % of the time in 1000 simulations of 18 players' batting averages and then 45 at bats.

R Code

1.
```
x <- read.csv("SteinData.csv", head = T)
#head(x)

phat.x <- mean(x$avg45)
sigma2.x <- phat.x*(1-phat.x)/45
c.x <- 1-((dim(x)[1] - 3)*sigma2.x/sum((x$avg45 - phat.x)^2))

z.x <- phat.x + c.x*(x$avg45 - phat.x)


msez.x <- sum((z.x - x$avgSeason)^2)
```

```
mse.x <- sum((x$avg45 - x$avgSeason)^2)
#mse.x
```

2. (a)
```
set.seed(53228)
n <- 18
rand_avgs <- runif(n,0.15,0.35)
```

(b)
```
m <- 45

rand_45 <- sapply(rand_avgs, function(t){rbinom(45,1,t)})

rand_mean_45 <- apply(rand_45, 2, mean)
```

(c)
```
phat_rand <- mean(rand_45)
sigma2_rand45 <- phat_rand*(1-phat_rand)/m
c_rand <- 1-((dim(rand_45)[2] - 3)*sigma2_rand45/sum((rand_mean_45 - phat_rand)^2))

z_rand <- phat_rand + c_rand*(rand_mean_45 - phat_rand)


mse_rand_mean_45 <- sum((rand_mean_45 - x$avgSeason)^2)
mse_z_rand <- sum((z_rand - x$avgSeason)^2)
```

(d)
```
n <- 18
m <- 45


rand_avgs_new <- data.frame(replicate(1000, runif(n,0.15,0.35)))


fn.b <- function(b){
rand_45 <- sapply(b, function(t){rbinom(45,1,t)})

rand_mean_45 <- apply(rand_45, 2, mean)

phat_rand <- mean(rand_45)
sigma2_rand45 <- phat_rand*(1-phat_rand)/m
c_rand <- 1-((dim(rand_45)[2] - 3)*sigma2_rand45/sum((rand_mean_45 - phat_rand)^2))

z_rand <- phat_rand + c_rand*(rand_mean_45 - phat_rand)


mse_rand_mean_45 <- sum((rand_mean_45 - x$avgSeason)^2)
mse_z_rand <- sum((z_rand - x$avgSeason)^2)

return(ifelse(mse_z_rand < mse_rand_mean_45, "Stein better", ifelse(mse_z_rand > mse_rand_mean_45, "Obs. better", "Inconclusive")))
}



best <- apply(rand_avgs_new, 2, fn.b)
prop.stein <- length(which(best == "Stein better"))/length(best)
```