STAT 206 Lab 4

Xin Feng(Vanessa)

10/25/2019

Due Monday, October 28, 5:00 PM

General instructions for labs: Labs must be completed as a pdf file. Give the commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file. Each answer must be supported by written statements as well as any code used.

Agenda: Distributions as models, method of moments and maximum likelihood estimation.

The Beta is a random variable bounded between 0 and 1 and often used to model the distribution of proportions. The probability distribution function for the Beta with parameters α and β is

$$p(x|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)+\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}$$

where $\Gamma()$ is the Gamma function, the generalized version of the factorial. Thankfully, for this assignment, you need not know what the Gamma function is; you need only know that the mean of a Beta is $\frac{\alpha}{\alpha+\beta}$ and its variance is $\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.

For this assignment you will test the fit of the Beta distribution to the on-base percentages (OBPs) of hitters in the 2014 Major League Baseball season; each plate appearance (PA) results in the batter reaching base or not, and this measure is the fraction of successful attempts. This set has been pre-processed to remove those players with an insufficient number of opportunities for success.

Part I

1. Load the file [http://faculty.ucr.edu/~jflegal/206/mlb-obp.csv] into a variable of your choice in R. How many players have been included? What is the minimum number of plate appearances required to appear on this list? Who had the most plate appearances? What are the minimum, maximum and mean OBP?

```r
mlb_obp <- read.csv("http://faculty.ucr.edu/~jflegal/206/mlb-obp.csv")
# View(mlb_obp)
```

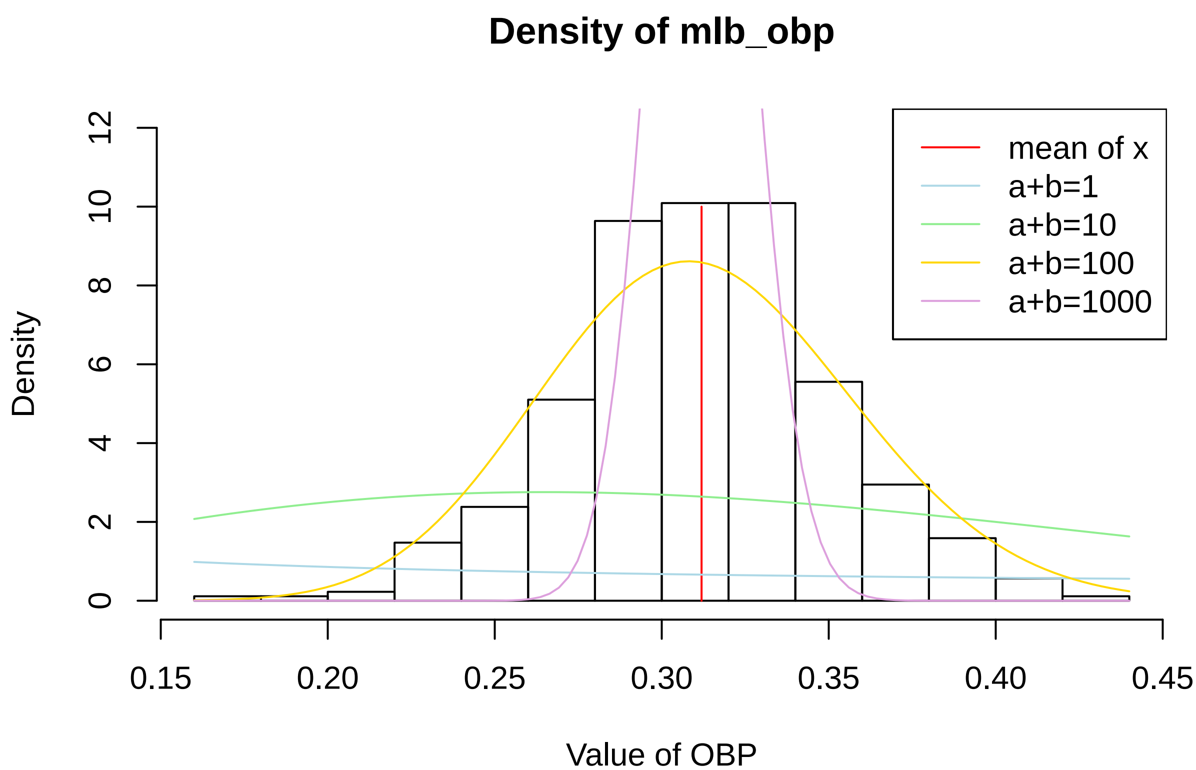
2. Plot the data as a histogram with the option `probability=TRUE`. Add a vertical line for the mean of the distribution. Does the mean coincide with the mode of the distribution?
3. Eyeball fit. Add a `curve()` to the plot using the density function `dbeta()`. Pick parameters α and β that matches the mean of the distribution but where their sum equals 1. Add three more `curve()`s to this plot where the sum of these parameters equals 10, 100 and 1000 respectively. Which of these is closest to the observed distribution?

```r
# 2: histogram
obp_mean <- mean(mlb_obp$OBP)
obp_var <- var(mlb_obp$OBP)
hist(mlb_obp$OBP, probability = TRUE, main="Density of mlb_obp", xlab="Value of OBP",
     ylim=c(0,12))
segments(obp_mean, 0, obp_mean, 10, col=2)
```

```
# 3: add curve
# According to the function of mean(x)=alpha/(alpha+beta), alpha+beta=1
# calculating the parameters of alpha and beta.
obp_mean
```

```
## [1] 0.3119184
```

```
obp_alpha <- obp_mean
obp_beta <- 1-obp_alpha
curve(dbeta(x,obp_alpha, obp_beta, ncp=0, log=FALSE), add=TRUE, col="lightblue")
obp_alpha <- 10*obp_mean
obp_beta <- 10-obp_alpha
curve(dbeta(x,obp_alpha, obp_beta, ncp=0, log=FALSE), add=TRUE, col="lightgreen")
obp_alpha <- 100*obp_mean
obp_beta <- 100-obp_alpha
curve(dbeta(x,obp_alpha, obp_beta, ncp=0, log=FALSE), add=TRUE, col="gold")
obp_alpha <- 1000*obp_mean
obp_beta <- 1000-obp_alpha
curve(dbeta(x,obp_alpha, obp_beta, ncp=0, log=FALSE), add=TRUE, col="plum")
legend("topright", legend=c("mean of x", "a+b=1", "a+b=10", "a+b=100", "a+b=1000"),
       col=c("red","lightblue","lightgreen","gold","plum"), lty=c(1,1,1,1,1))
```

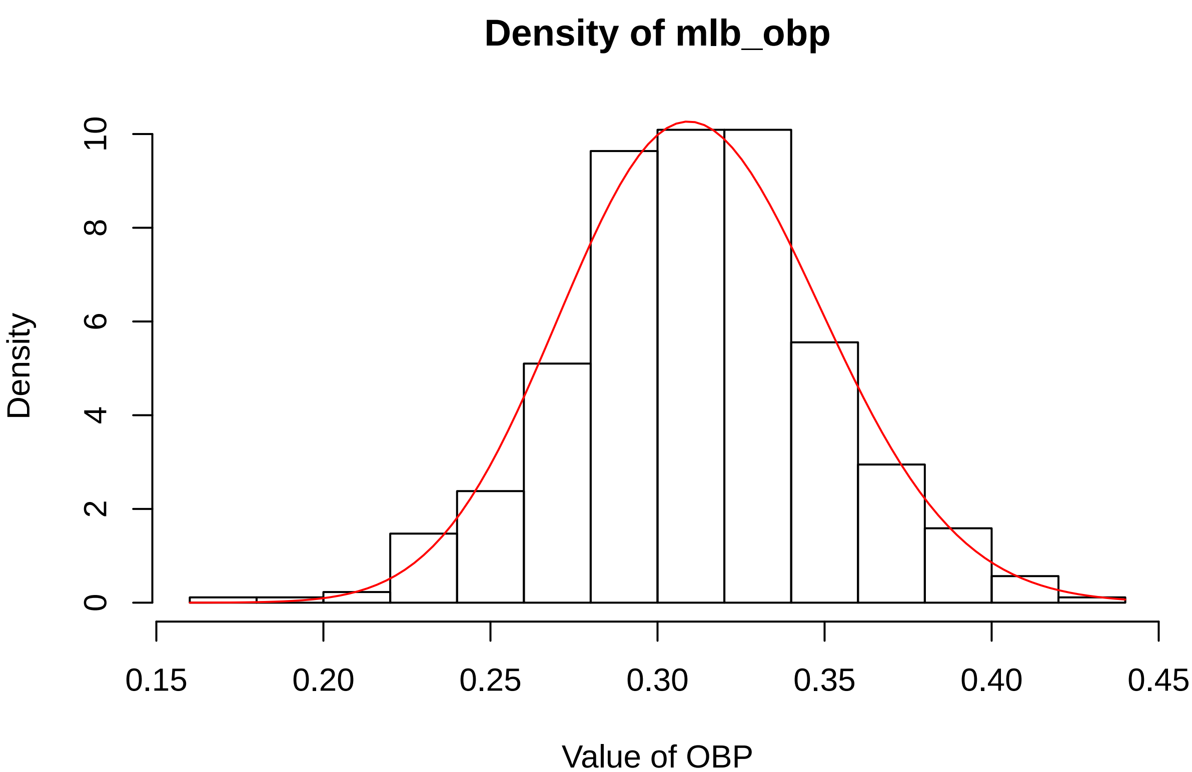



```
# According to the picture, when alpha+beta=100, the Beta curve is closest to the observed
# distribution.
```

Part I

4. Method of moments fit. Find the calculation for the parameters from the mean and variance from [http://en.wikipedia.org/wiki/Beta_distribution] and solve for α and β. Create a new density histogram and add this `curve()` to the plot. How does it agree with the data?

```
# According to the function: Mean=a/(a+b), Var=ab/(a+b)^2*(a+b+1),
# calculate that a=(Mean^2-Mean^3-Mean*Var)/Var, b=(1-Mean)*a/Mean
obp_alpha <- (obp_mean^2-obp_mean^3-obp_mean*obp_var)/obp_var
obp_beta <- (1-obp_mean)*obp_alpha/obp_mean
hist(mlb_obp$OBP, probability = TRUE, main="Density of mlb_obp", xlab="Value of OBP")
fun_beta <- function(x) dbeta(x,obp_alpha, obp_beta, ncp=0, log=FALSE)
curve(fun_beta, add=TRUE, col=2)
```



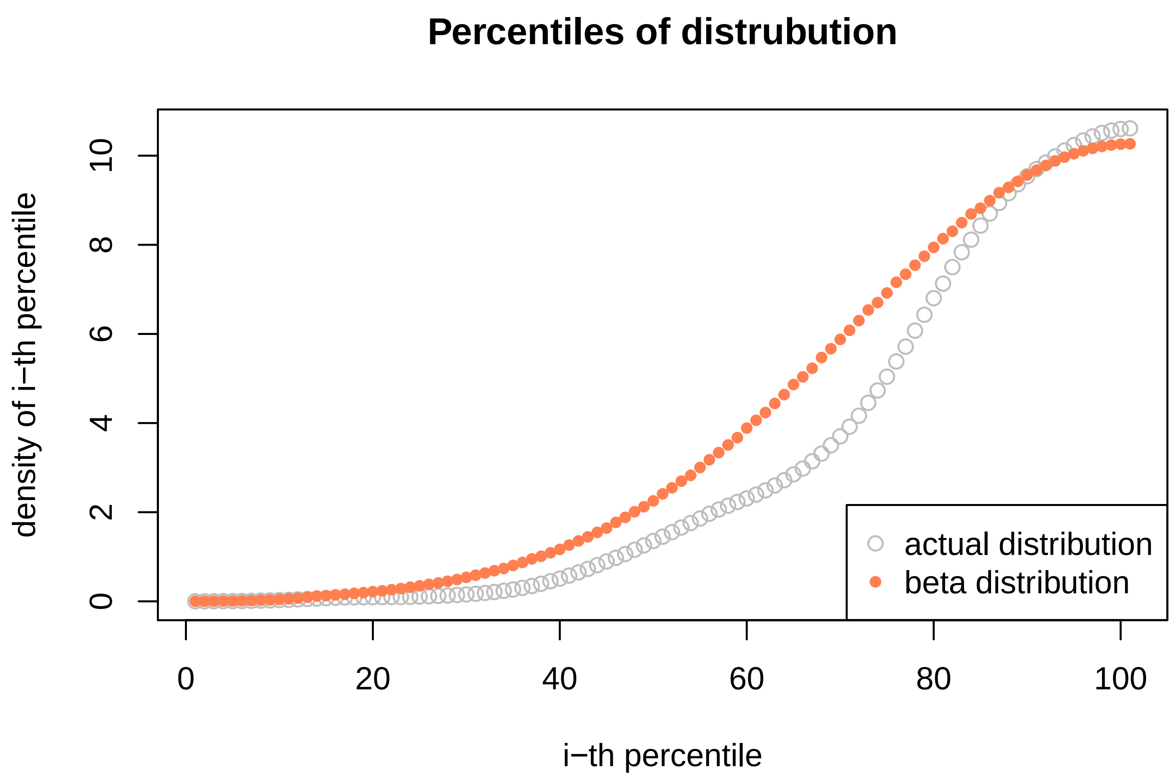

```
# The Beta distribution is very closed to the observed distribution.
```

5. Calibration. For the previous part, find the 99 percentiles of the actual distribution using the `quantile()` function and plot them against the 99 percentiles of the beta distribution you just fit using `qbeta()`. How does the fit appear to you?

```
data_x <- density(mlb_obp$OBP)
obp_data_qt <- unname(quantile(data_x[[2]], seq(0, 1, 0.01)))
beta_x <- seq(min(mlb_obp$OBP), max(mlb_obp$OBP), 1e-3)
obp_beta_qt <- unname(quantile(fun_beta(beta_x), seq(0, 1, 0.01)))

plot(obp_data_qt, col="grey", pch=1, main="Percentiles of distrubution",
     xlab="i-th percentile", ylab="density of i-th percentile")
```

```r
points(obp_beta_qt, col="coral", pch=20)
legend("bottomright", legend=c("actual distribution", "beta distribution"),
       col=c("grey","coral"), pch=c(1,20))
```




```r
# According to the picture, percentiles of the two distributions is closed,
# so the fitting is good.
```

6. Create a function for the log-likelihood of the distribution that calculates `-sum(dbeta(your.data.here, your.alpha, your.beta, log=TRUE))` and has one argument `p=c(your.alpha, your.beta)`. Use `nlm()` to find the minimum of the negative of the log-likelihood. Take the MOM fit for your starting position. How do these values compare?

```r
obp_loglh <- function(p,x){
  alpha <- p[1]
  beta <- p[2]
  n <- length(x)
  func_b <- function(alpha, beta,x) dgamma(x, alpha)*dgamma(x, beta)/dgamma(x, alpha+beta)
  logL <- -sum(dbeta(x, alpha, beta, log=TRUE))
  return(logL)
}
result <- nlm(obp_loglh, c(obp_alpha, obp_beta), mlb_obp$OBP)
result
```

```
## $minimum
## [1] -805.8799
##
## $estimate
```

```
## [1] 43.73915 96.49892
##
## $gradient
## [1] 1.499739e-06 4.461522e-06
##
## $code
## [1] 1
##
## $iterations
## [1] 4
```

```r
# The minimum of the negative of the log-likelihood is the maximum log-likelihood.
cat("Initial alpha: ", obp_alpha, ", Estimate of alpha: ", result$estimate[1], "\n")
```

```
## Initial alpha:  44.3169 , Estimate of alpha:  43.73915
```

```r
cat("Initial alpha: ", obp_beta, ", Estimate of alpha: ", result$estimate[2], "\n")
```

```
## Initial alpha:  97.76163 , Estimate of alpha:  96.49892
```