# Lab 6

*YI CHEN (YC3356)*

*November 14, 2017*

# Instructions

Before you leave lab today make sure that you upload a .pdf file to the canvas page (this should have a .pdf extension). This should be the PDF output after you have knitted the file, we don't need the .Rmd file (don't upload the one with the .Rmd extension). The file you upload to the Canvas page should be updated with commands you provide to answer each of the questions below. You can edit this file directly to produce your final solutions. Note, however, in the file you upload you should the above header to have the date, your name, and your UNI. Similarly, when you save the file you should replace **UNI** with your actualy UNI.

# Background

In today's lab we will use the Beta distribution to explore the probability of reaching a base safely in baseball. 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 $\alpha$ and $\beta$ 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 1

1. Load the file `baseball.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?

```
setwd("C:/Users/cheny/Desktop/study/statistical computing and intro to data science/lab/lab s
ix")
baseball <- read.csv('baseball.csv',header = TRUE)

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
baseball %>% summarise(number_of_players=n_distinct(Name),min_of_plate=min(PA),max_of_plate=m
ax(PA),min_OBP=min(OBP),max_OBP=max(OBP),mean_OBP=mean(OBP))
```

```
##   number_of_players min_of_plate max_of_plate min_OBP max_OBP  mean_OBP
## 1               441          103          726   0.168   0.432 0.3119184
```

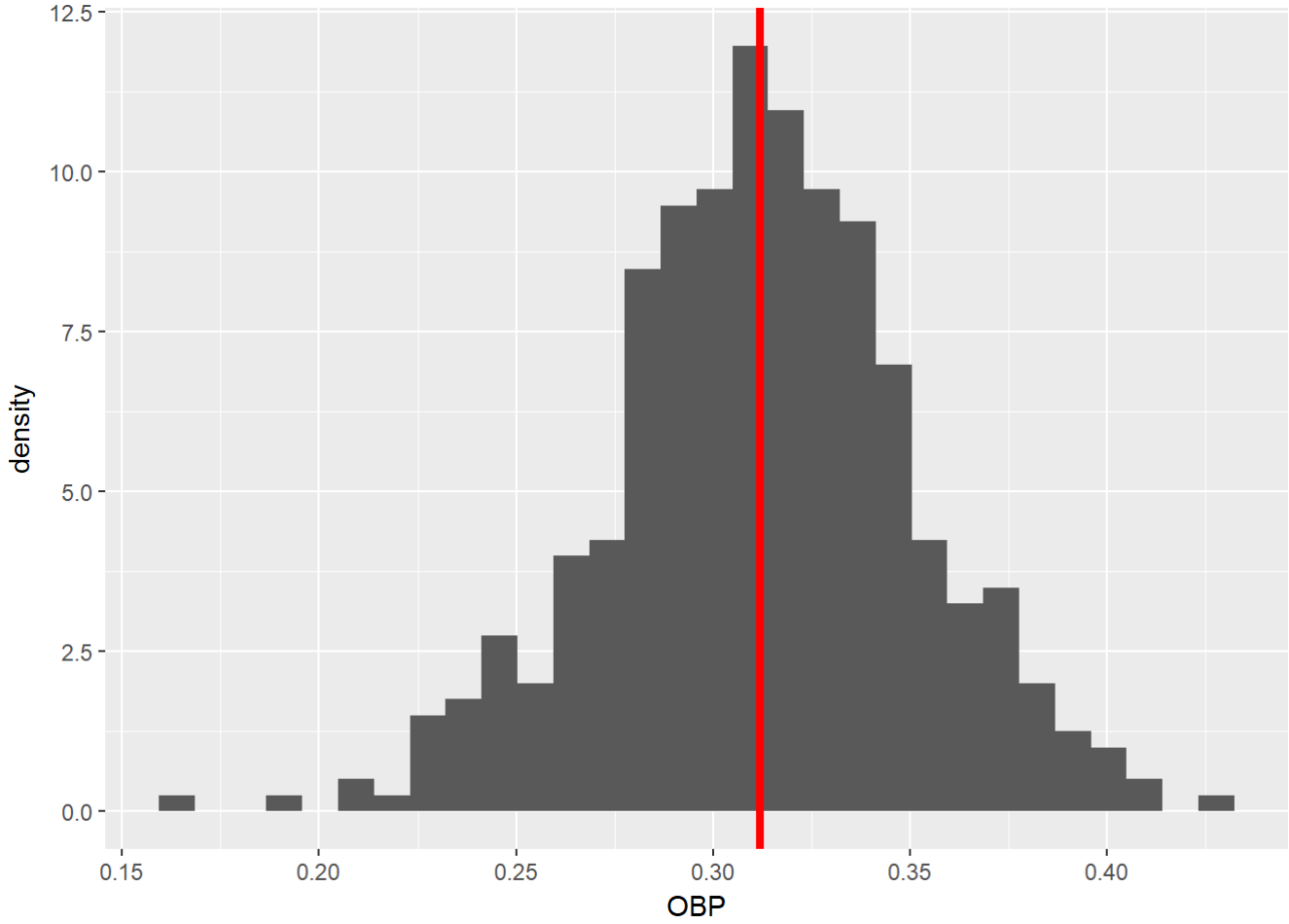
2. Plot the `OBP` data as a histogram with the aesthetic `y = ..density..`. Add a vertical line for the mean of the distribution. Does the mean coincide with the mode of the distribution?

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
ggplot(data = baseball)+
        geom_histogram(aes(x=OBP,y=..density..))+
        geom_vline(xintercept=mean(baseball$OBP),lwd=1.5,col='red')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

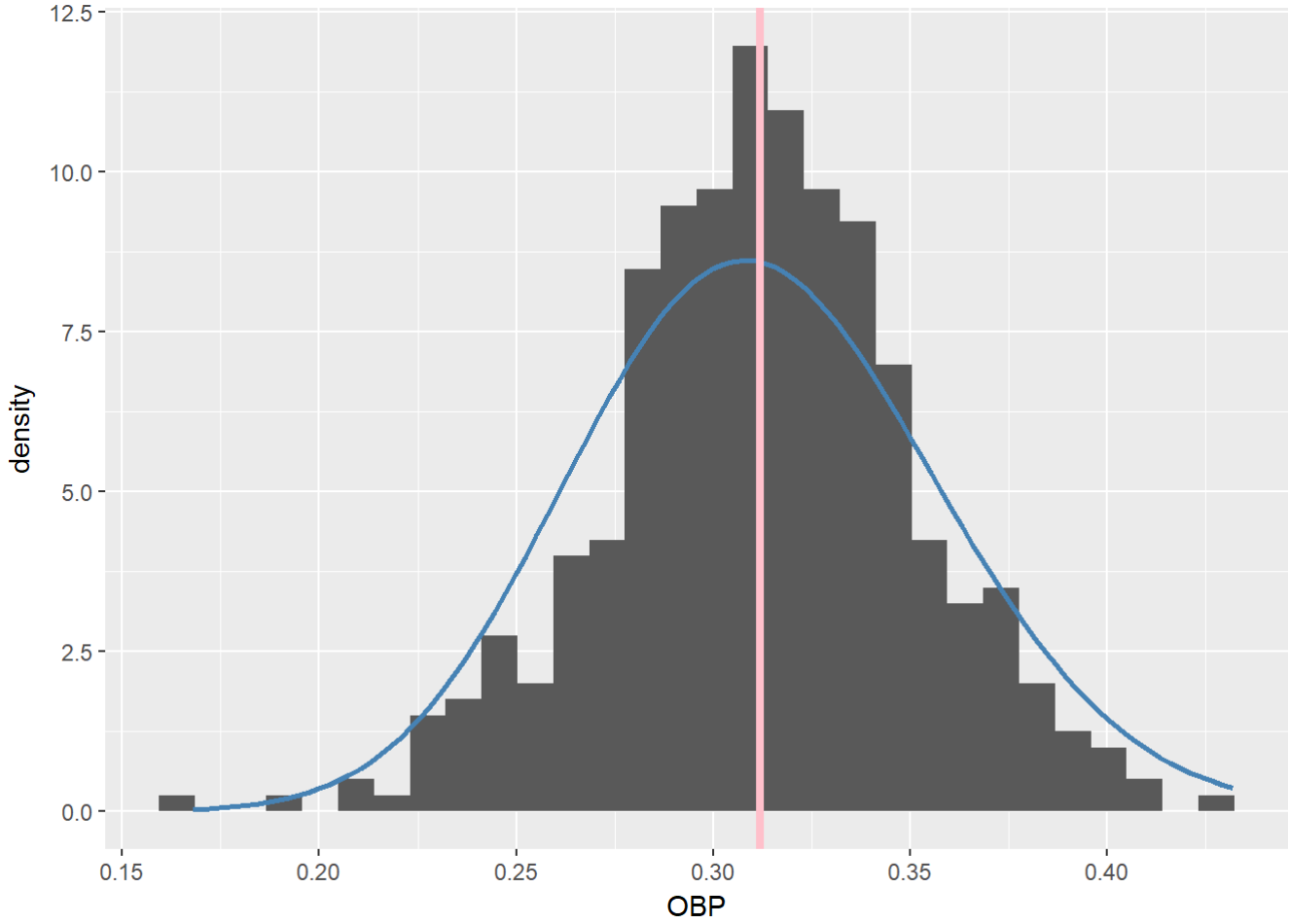



3. Eyeball Fit. Now add a Beta density function to the previous plot using `stat_function()` and the density function `dbeta()`. Let `this.mean` be a variable assigned to be the mean of the `OBP` variable. Pick parameters $\alpha = 100 * this.mean$ and $\beta = 100 - \alpha$.

```
this.mean <- mean(baseball$OBP)
alpha <- 100*this.mean
beta <- 100-alpha

ggplot(data = baseball)+
        geom_histogram(aes(x=OBP,y=..density..))+
        stat_function(aes(x=OBP),fun = dbeta,args = list(alpha,beta),lwd=1,col='steelblue')+
        geom_vline(xintercept=mean(baseball$OBP),lwd=1.5,col='pink')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




# Part 2

(Note that at this point in the semester, Part 1 should be easy. There are problems if you are unable to do basic exploratory data analysis and plotting.)

4. Method of moments fit. Using the values for the mean $\mu = \frac{\alpha}{\alpha+\beta}$ and variance $\sigma^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$, solve for $\alpha$ and $\beta$ (as functions of the mean and variance). Use these to find the Methods of Moments estimates. With the new estimates, create a new density histogram and add this Method of Moments fit to the plot. How does it agree with the data?

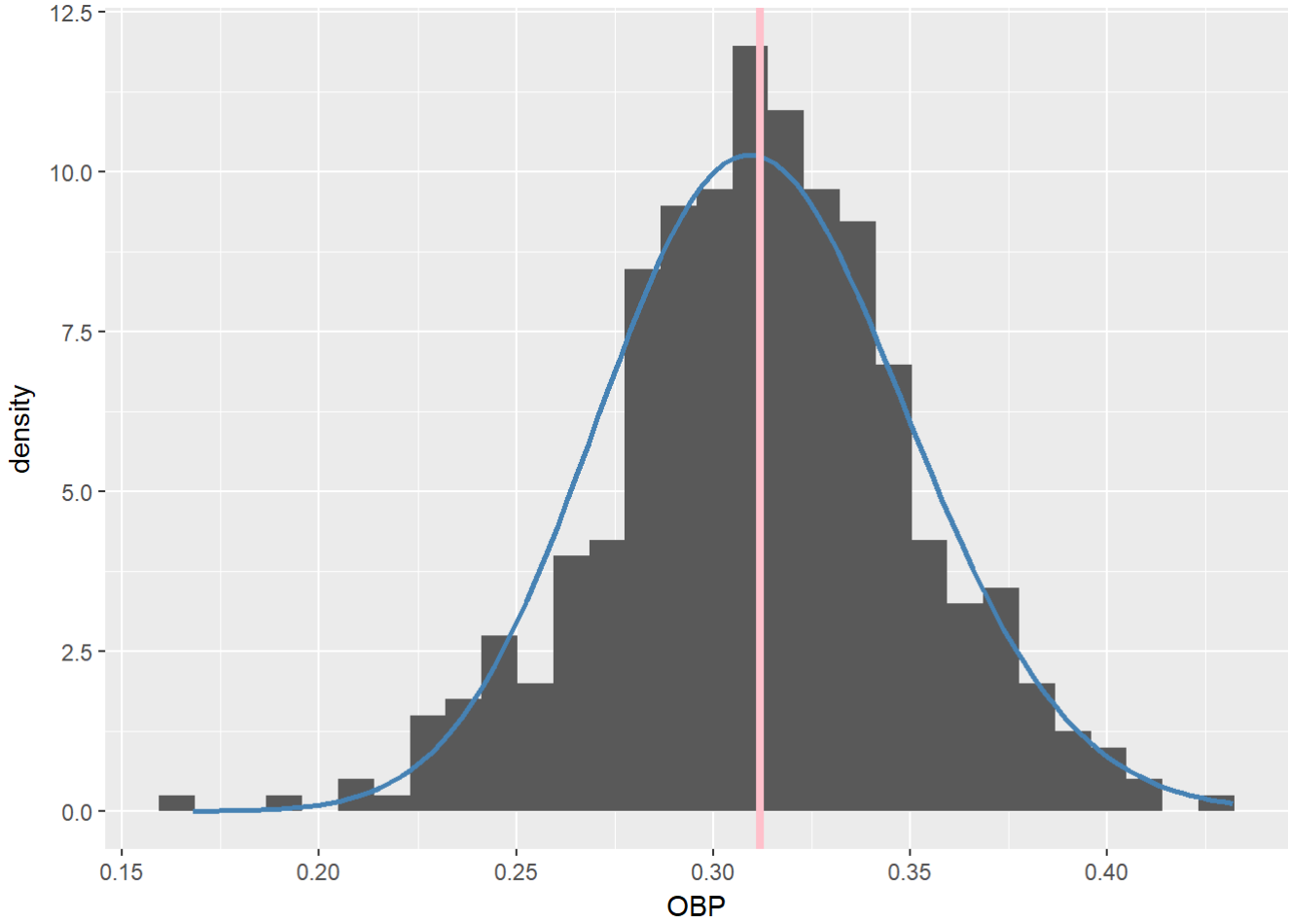
```
m <- mean(baseball$OBP)
v <- sd(baseball$OBP)

gam.alpha <- ((m/v^2)*(m*(1-m)-v^2))

gam.beta <- ((1-m)/v^2)*(m*(1-m)-v^2)

ggplot(data = baseball)+
        geom_histogram(aes(x=OBP,y=..density..))+
        stat_function(aes(x=OBP),fun = dbeta,args = list(gam.alpha,gam.beta),lwd=1,col='steel
blue')+
        geom_vline(xintercept=mean(baseball$OBP),lwd=1.5,col='pink')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



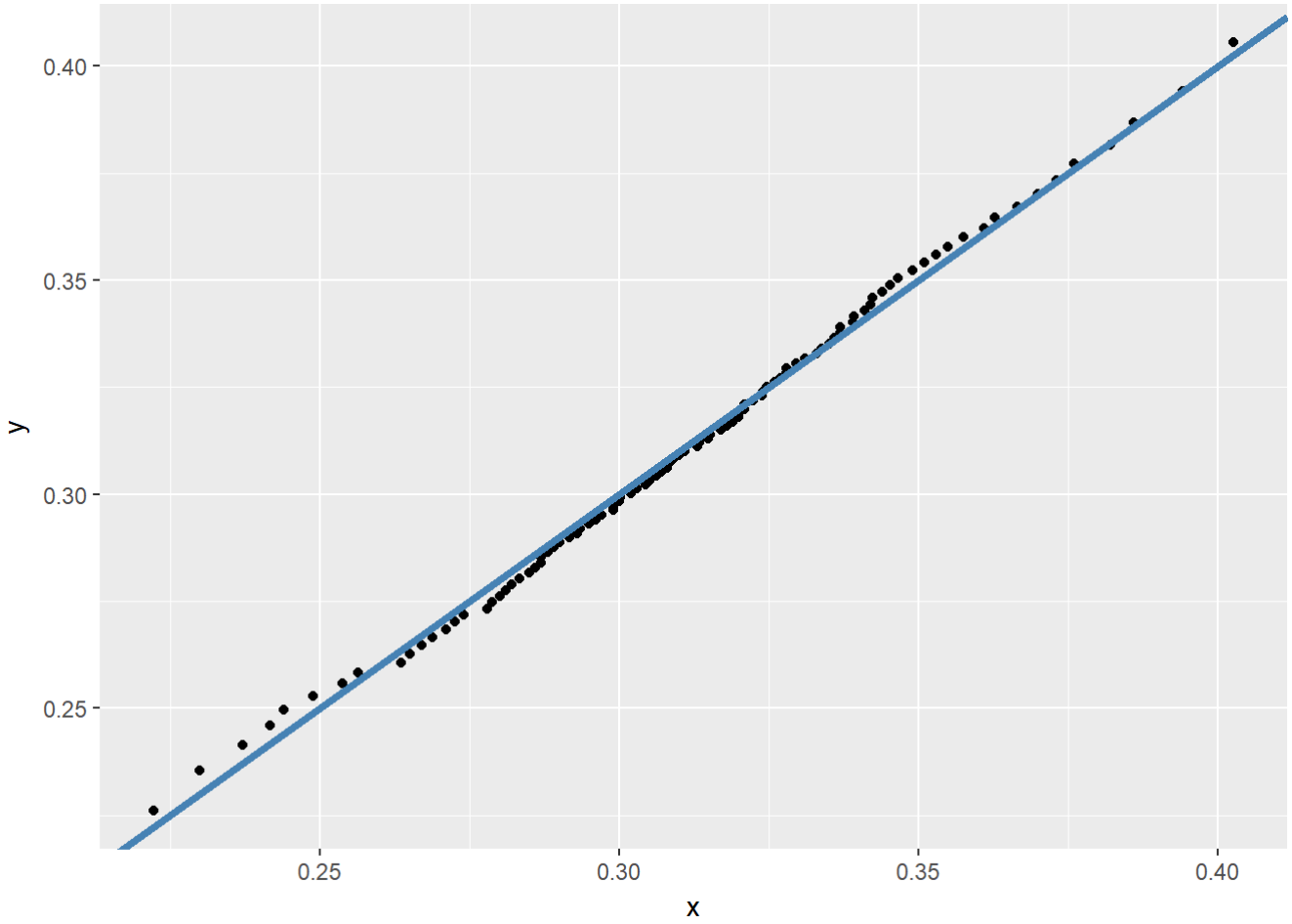

Hint: Solving for $\alpha$ and $\beta$ you should find

$$\alpha = \frac{\mu}{\sigma^2}\left[\mu(1-\mu) - \sigma^2\right], \quad \text{and} \quad \beta = \frac{1-\mu}{\sigma^2}\left[\mu(1-\mu) - \sigma^2\right].$$

How? Solve the mean equation for $\beta$ and then plug this value of $\beta$ into the variance equation.

5. Calibration. Find the 99 percentiles of the actual distribution of the data using the `quantile()` function using `quantile(bb$OBP, probs = seq(1, 99)/100)` and plot them against the 99 percentiles of the beta distribution you just fit using `qbeta()`. How does the fit appear to you?

```
x <- quantile(baseball$OBP,probs = seq(1,99)/100)
y <- qbeta(p=seq(1,99)/100,shape1 = gam.alpha,shape2 = gam.beta)
data <- data.frame(x=x,y=y)
ggplot(data = data)+
        geom_point(aes(x=x,y=y))+
        geom_abline(slope = 1,intercept = 0,lwd=1.5,col='steelblue')
```




6. Optional if you have time – MLE fit. Create a function that calculates the (negative) log-likelihood of the distribution. Hint: Calculate this value with code like `-sum(dbeta(your.data.here, your.alpha, your.beta, log = TRUE))`. The function should have argument `params = c(your.alpha, your.beta)`. Find the minimum of the negative of the log-likelihood using the optimization function `nlm()`. Take the Method of Moments fit for your starting position. How do these values compare?