STATS 205: Homework Assignment 3

Brian Liu

4/26/2019

Solution to Problem 1

(i)

```
library(bootstrap); data(law)
t(law)
```

```
##          1     2      3      4      5      6   7      8      9     10
## LSAT 576.00 635.0 558.00 578.00 666.00 580.00 555 661.00 651.00 605.00
## GPA    3.39   3.3   2.81   3.03   3.44   3.07   3   3.43   3.36   3.13
##          11     12     13     14     15
## LSAT 653.00 575.00 545.00 572.00 594.00
## GPA    3.12   2.74   2.76   2.88   2.96
```

```
theta.hat = cor(law$LSAT, law$GPA); theta.hat
```

```
## [1] 0.7763745
```

```
library(partitions)
n = 15
allCompositions = compositions(n, n);allCompositions[,1:5]
```

```
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]   15   14   13   12   11
##  [2,]    0    1    2    3    4
##  [3,]    0    0    0    0    0
##  [4,]    0    0    0    0    0
##  [5,]    0    0    0    0    0
##  [6,]    0    0    0    0    0
##  [7,]    0    0    0    0    0
##  [8,]    0    0    0    0    0
##  [9,]    0    0    0    0    0
## [10,]    0    0    0    0    0
## [11,]    0    0    0    0    0
## [12,]    0    0    0    0    0
## [13,]    0    0    0    0    0
## [14,]    0    0    0    0    0
## [15,]    0    0    0    0    0
```

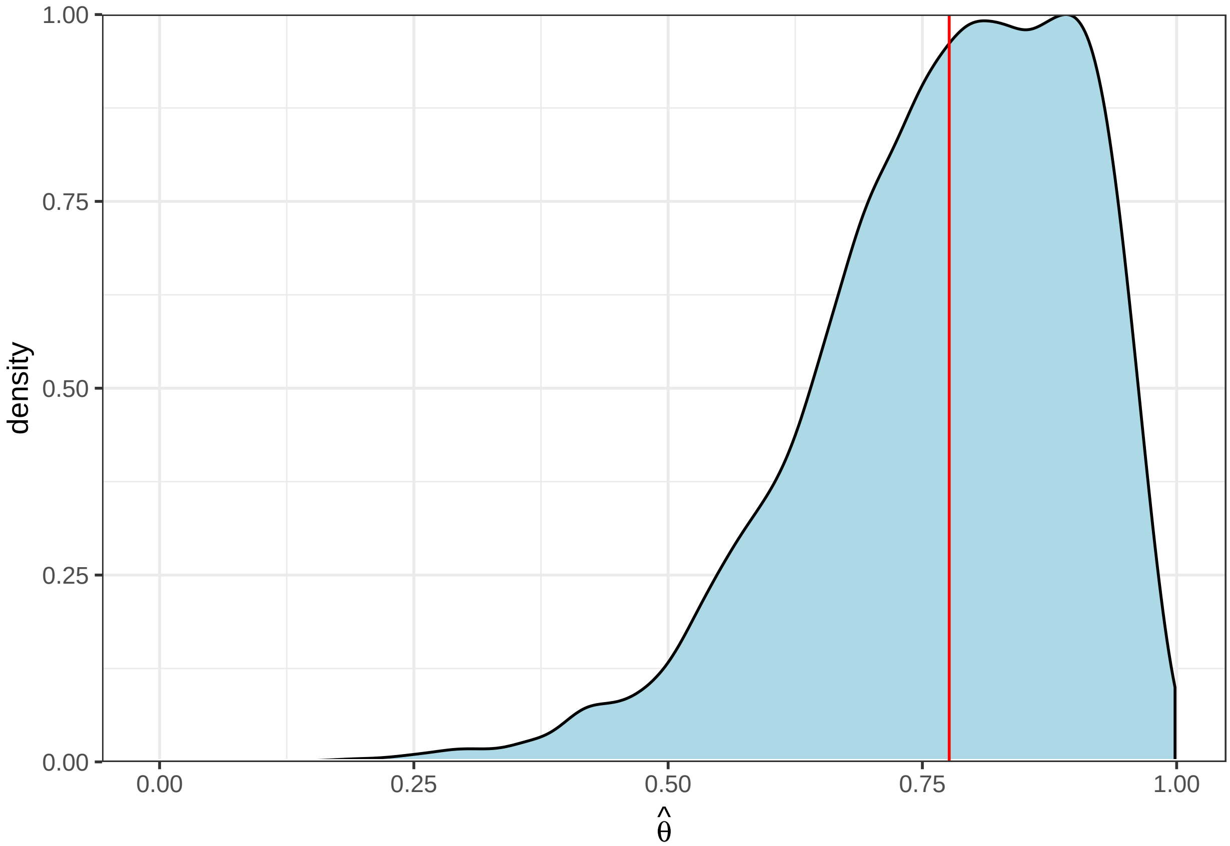
```
allCompositions.sub = allCompositions[, sample(1:dim(allCompositions)[2], size=10000, replace=FALSE)]

draw.bootstrap.samples = function(df){
  n = dim(df)[1]
  ind = sample(n, replace = TRUE)
  cor.bootstrap.replicate = cor(df[ind, "LSAT"], df[ind, "GPA"])
  return(cor.bootstrap.replicate)
}
R = 10000
```

```r
theta.hat.star = replicate(R, draw.bootstrap.samples(law))
# make a ggplot
library(ggplot2)
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```r
theta.hat.star.df = data.frame(theta.hat.star = theta.hat.star)
ggplot(theta.hat.star.df) +
  geom_density(aes(x = theta.hat.star, y = ..scaled..),
    fill = "lightblue") +
  geom_hline(yintercept=0, colour="white", size=1) +
  theme_bw() +
  ylab("density") +
  xlab(bquote(hat(theta))) +
  geom_vline(xintercept = theta.hat, col = "red")+
  scale_y_continuous(expand = c(0,0))
```



(ii)

```r
sd(theta.hat.star)
```

```
## [1] 0.133338
```

Solution to Problem 2

(i)

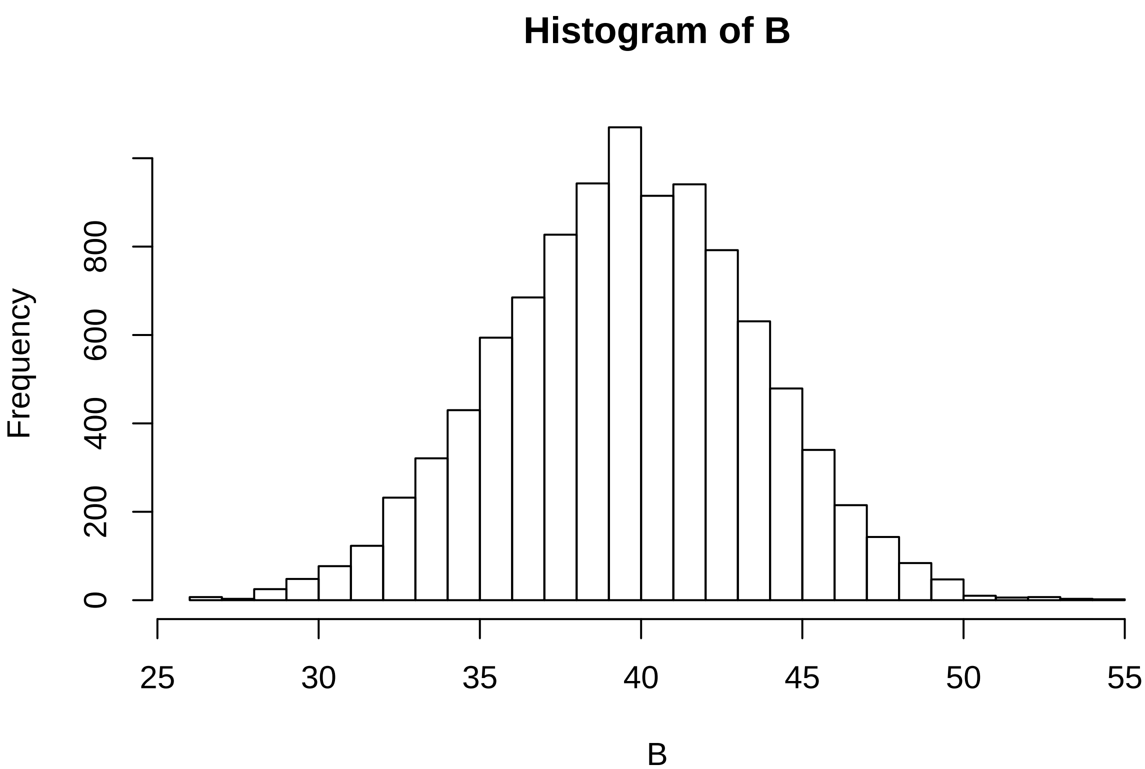
```
67 runs resulting in swallowing attempts
58 successful
9 failed

H_0 : p = 0.6
H_A : p > 0.6
```

```r
n = 67
successes = 58
pbar = successes / n; pbar
```

```
## [1] 0.8656716
```

```r
p0 = 0.6; nsim = 10000
B = rbinom(nsim, size = n, prob = p0)
hist(B, breaks = 30)
```



Test statistic Z:

$$Z_0 = \frac{B - 67(0.6)}{(67(0.6)(0.4))^{\frac{1}{2}}}$$

```r
qnorm((1-0.05), mean = 0, sd = 1)
```

```
## [1] 1.644854
```

Rejection region: $Z \geq z_{0.05} = 1.645$

Observed test statistic Z_o:

$$Z_o = \frac{58 - 67(0.6)}{(67(0.6)(0.4))^{\frac{1}{2}}} = 4.44$$

```
numerator = successes - (n * p0)
denominator = sqrt(n * p0 * (1.0 - p0))
Z.obs = numerator / denominator; Z.obs
```

```
## [1] 4.438917
```

The large sample approximation value $Z_o = 2.5 > 1.645$ and thus we reject $H_0 : p = 0.6$ in favor of $p > 0.6$ at the approximate $\alpha = 0.05$ level. Thus there is evidence that the success rate of swallowing attempts is greater than 0.6.

(ii)

Power is the probability of rejecting H_0 when H_A is true. We found that test reject H_0 is $Z \geq z_{0.05} = 1.645$. Therefore, if $p = 0.7$,

$$Z_o = \frac{58 - 67(0.6)}{(67(0.6)(0.4))^{\frac{1}{2}}} = 4.44$$

is no longer standard normal.

We have

$$Z_{o7} = \frac{58 - 67(0.7)}{(67(0.7)(0.3))^{\frac{1}{2}}} = 2.96$$

```
p1 = 0.7
numerator = successes - (n * p1)
denominator = sqrt(n * p1 * (1.0 - p1))
Z.obs.seven = numerator / denominator; Z.obs.seven
```

```
## [1] 2.959211
```

$$Power = P(Z \geq 1.645 | p = 0.7)$$

$$= P_{p=0.7}(\frac{B - 67(0.6)}{(67(0.6)(0.4))^{\frac{1}{2}}} \geq 1.645)$$

$$= P_{p=0.7}(B \geq 1.645(67(0.6)(0.4))^{\frac{1}{2}} + 67(0.6))$$

$$= P_{p=0.7}(\frac{B - 67(0.7)}{(67(0.7)(0.3))^{\frac{1}{2}}} \geq \frac{1.645(67(0.6)(0.4))^{\frac{1}{2}} + 67(0.6)}{)}$$

There is a function in R that calculates the same thing:

```
library(pwr)
pwr.p.test(h = 0, n = 67, sig.level = 0.05, power = NULL, alternative = c("greater"))
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0
##               n = 67
##       sig.level = 0.05
##           power = 0.05
##     alternative = greater
```

```r
library(beepr)
beep()
```