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
```

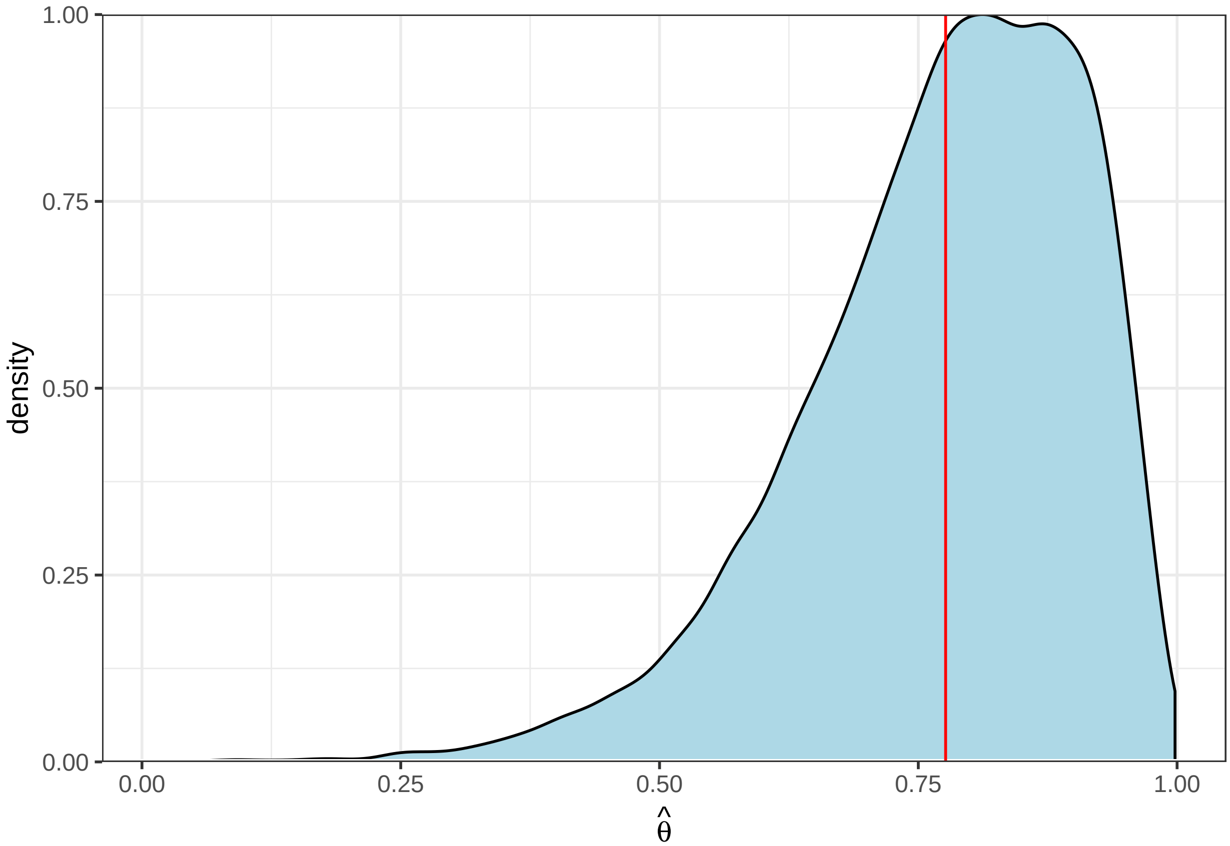
```
allCompositions.sub = allCompositions[, sample(1:dim(allCompositions)[2], size=10000, replace=FALSE)]

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
```

```
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
```

```
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)

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

```
## [1] 0.1351167
```

Solution to Problem 2

(i)

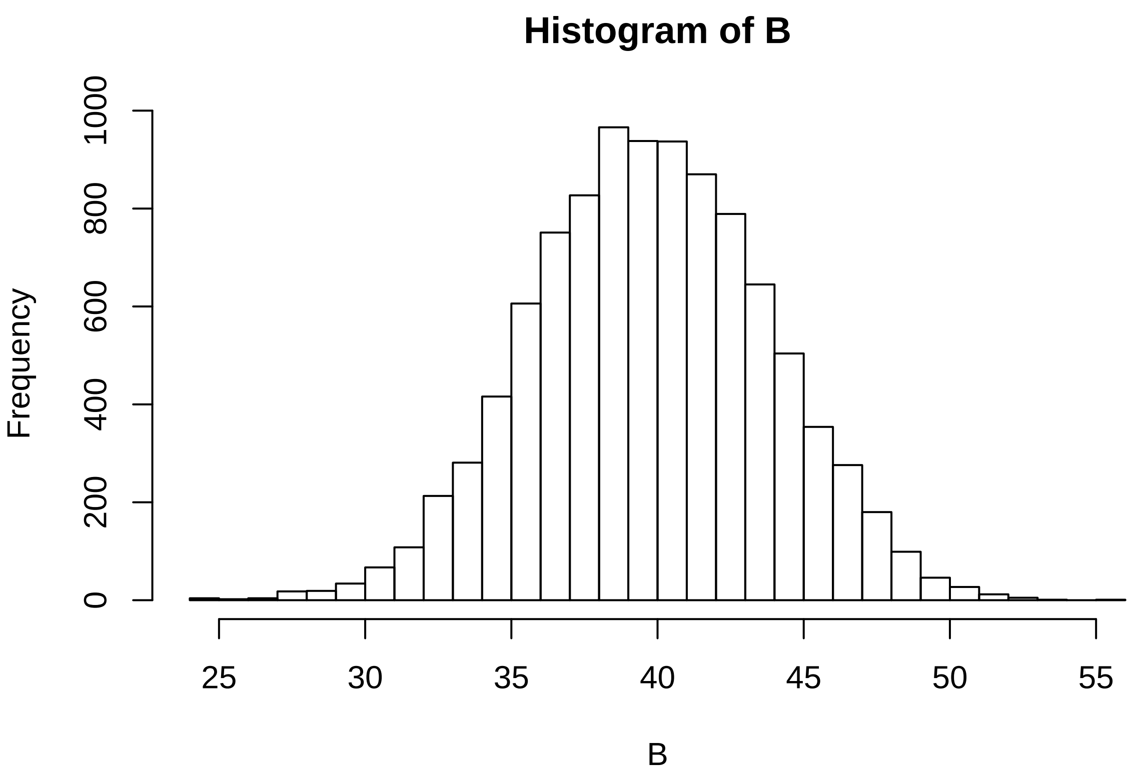
```
67 runs resulting in swallowing attempts
58 successful
9 failed

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
```

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

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

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

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

```
triple_product = n * p0 * (1.0 - p0)
first_term = 1.645 * sqrt(triple_product)
second_term = n * p0
third_term = n * p1
bottom_term = n * p1 * (1.0 - p1)
```

```
p7_numerator = first_term + second_term - third_term
p7_denominator = sqrt(bottom_term)
Pp_7_zvalue = p7_numerator / p7_denominator; Pp_7_zvalue
```

```
## [1] -0.02761144
```

$$P(Z^* \geq -0.0276) = 0.4890$$

```
# pvalue = pnorm(-abs(Pp_7_zvalue)); pvalue
pvalue = pnorm(Pp_7_zvalue); pvalue
```

```
## [1] 0.488986
```

If $p = 0.8$,

$$Power = P(Z \geq 1.645 | p = 0.8)$$

$$= P_{p=0.8}\left(\frac{B - 67(0.8)}{(67(0.8)(0.2))^{\frac{1}{2}}} \geq \frac{1.645(67(0.6)(0.4))^{\frac{1}{2}} + 67(0.6) - 67(0.8)}{(67(0.8)(0.2))^{\frac{1}{2}}}\right)$$

```
p2 = 0.8
triple_product = n * p0 * (1.0 - p0)
first_term = 1.645 * sqrt(triple_product)
second_term = n * p0
third_term = n * p2
bottom_term = n * p2 * (1.0 - p2)
p8_numerator = first_term + second_term - third_term
p8_denominator = sqrt(bottom_term)
Pp_8_zvalue = p8_numerator / p8_denominator; Pp_8_zvalue
```

```
## [1] -2.077971
```

$$P(Z^* \geq -2.078) = 0.01886$$

```
# pvalue = pnorm(-abs(Pp_7_zvalue)); pvalue
pvalue = pnorm(Pp_8_zvalue); pvalue
```

```
## [1] 0.01885601
```

Solution to Problem 3

Summary: **Estimate for** $\hat{p} = 0.8615$ and **estimate for standard deviation of** $\hat{p} = 0.04284$.

Estimate for p using binomial confidence interval, `binom.confint()`:

```
library(binom)
binom.confint(x=56, n=65, conf.level=.95, methods = "asymptotic")
```

```
##       method  x  n      mean     lower     upper
## 1 asymptotic 56 65 0.8615385 0.7775744 0.9455025
```

$$\hat{p} = (0.7776, 0.9455)$$

Estimate for p using 1-sample proportions test without continuity correction, `prop.test()`:

```
prop.test(x=56, n=65, p = 0.6, conf.level=0.95, alternative = c("greater"))
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  56 out of 65, null probability 0.6
## X-squared = 17.452, df = 1, p-value = 1.473e-05
## alternative hypothesis: true p is greater than 0.6
## 95 percent confidence interval:
##  0.7676875 1.0000000
## sample estimates:
##         p
## 0.8615385
```

$$p = 0.8615$$

Estimate for p using Exact Binomial Test:

```
binom.test(x=56, n=65, p = 0.6, alternative = c("greater"), conf.level = 0.95)
```

```
##
##  Exact binomial test
##
## data:  56 and 65
## number of successes = 56, number of trials = 65, p-value =
## 4.096e-06
## alternative hypothesis: true probability of success is greater than 0.6
## 95 percent confidence interval:
##  0.7708174 1.0000000
## sample estimates:
## probability of success
##              0.8615385
```

Standard error of $\hat{p}$ is:

$$\sqrt{\frac{p(1-p)}{n}}$$
$$= \sqrt{\frac{(0.6)(0.4)}{65}}$$
$$= 0.06076$$

```
p = 0.6
n = 65
numerator = p * (1 - p)
denominator = n
answer = sqrt(numerator/denominator); answer
```

```
## [1] 0.06076436
```

and estimate is:

$$\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$
$$= \sqrt{\frac{(0.8615)(1-0.8615)}{65}}$$
$$= 0.04284$$

```
p.hat = 0.8615
n = 65
numerator = p.hat * (1 - p.hat)
denominator = n
answer = sqrt(numerator/denominator); answer
```

```
## [1] 0.04284458
```

Solution to Problem 4

```
binom.confint(x = 56, n = 65, conf.level = 0.96, methods = "all")
```

```
##           method  x  n      mean     lower     upper
## 1  agresti-coull 56 65 0.8615385 0.7488973 0.9301180
## 2     asymptotic 56 65 0.8615385 0.7735567 0.9495202
## 3          bayes 56 65 0.8560606 0.7655984 0.9375798
## 4        cloglog 56 65 0.8615385 0.7439982 0.9276404
## 5          exact 56 65 0.8615385 0.7480632 0.9371740
## 6          logit 56 65 0.8615385 0.7484912 0.9286194
## 7         probit 56 65 0.8615385 0.7545841 0.9312980
## 8        profile 56 65 0.8615385 0.7589798 0.9334925
## 9            lrt 56 65 0.8615385 0.7589836 0.9335307
## 10     prop.test 56 65 0.8615385 0.7483484 0.9308913
## 11        wilson 56 65 0.8615385 0.7514483 0.9275670
```

```
# Wilson confidence interval
# binom.confint(x = 56, n = 65, conf.level = 0.96, methods = c("wilson"))

# Laplace-Wald confidence interval

# Agresti-Coull
# binom.confint(x = 56, n = 65, conf.level = 0.96, methods = c("agresti-coull"))

# Clopper-Pearson
# binom.lrt(x = 56, n = 65, conf.level = 0.96, conf.adj = FALSE)
```