Stat 111 Homework 2

Kojin Oshiba

1. Sampling from major league baseball player data

(a) population mean, std dev

```
calc_pop_sd <- function (vec) {
  sqrt(sum((vec-mean(vec))**2) / length(vec))
}

df          <- read.csv(file="baseball.csv", header=TRUE, sep=",")
K           <- nrow(df)
pop.mean    <- mean(df$Salary)
pop.stddev <- calc_pop_sd(df$Salary)
cat("population mean:", pop.mean, '\n')
```

```
## population mean: 1183417
```

```
cat("population standard deviation:", pop.stddev)
```

```
## population standard deviation: 1389991
```

(b) random sample

The population $(y_1, ..., y_K)$ is fixed. The sample $Y^* = (Y_1, ..., Y_5)'$ is random.

(c) std dev across sample averages

```
R = 100
calc_sample_avgs <- function(size) {
    replicate(R, {
                     sample.salaries <- sample(df$Salary,size=size,replace=T)
                     mean(sample.salaries)
                   })
}

cat("std dev across sample averages for n = 5:",sd(calc_sample_avgs(5)),'\n')
```

```
## std dev across sample averages for n = 5: 645278.5
```

```
cat("std dev across sample averages for n = 20:",sd(calc_sample_avgs(20)))
```

```
## std dev across sample averages for n = 20: 275025.4
```

The sample averages with $n = 20$ are better than those based on $n = 5$. This is because, by the law of large numbers, the standard deviation of a sample average as well as the stndard deviation across the 100 sample averages decrease. Hence, the sample averages with $n = 20$ are more likely to be a better proxy for μ.

(d)

i.

```
calc_sample_stddev_of_sample_avg <- function(size) {
  sample_avgs <- calc_sample_avgs(size)
  sqrt(sum((sample_avgs-mean(sample_avgs)) ** 2) / (R-1) )
}
cat("a sample standard deviation of a sample average using",'\n',
    "100 sample averages with sample size of each...",'\n')
```

```
## a sample standard deviation of a sample average using
##  100 sample averages with sample size of each...
```

```
cat("n=5: ", calc_sample_stddev_of_sample_avg(5),'\n')
```

```
## n=5:  639201.3
```

```
cat("n=20:", calc_sample_stddev_of_sample_avg(20),'\n')
```

```
## n=20: 323832.7
```

```
cat("n=80:", calc_sample_stddev_of_sample_avg(80))
```

```
## n=80: 148834.1
```

The term "this" was unclear whether it is "a proxy for the standard deviation of a sample average" or "the standard deviation of a sample average". From asking a TF, I assume that it is the latter. The latter is the square root of $Var(\bar{Y}^*)$. The relationship between this and σ is, from the lecture note,

$$Var(\bar{Y}^*) = \frac{1}{n}\sigma^2$$

ii.

salaries

iii.

sample mean salary

iv.

From the lecture note,

$$Var(\bar{Y}^*) = \frac{1}{n}\sigma^2$$

This is essentially the same answer as in i., but a TF said it's ok…

2. Bootstraping from a major league baseball player sample

(a)

```
n = 20
Y.star = sample(df$Salary,size=n,replace=T)
sample.sigma <- sd(Y.star)
cat("std dev of the sample mean (formulaic):",sample.sigma / sqrt(n))
```

```
## std dev of the sample mean (formulaic): 287215.3
```

σ should be approximated using the sample standard deviation for the 20 samples from the population. Then, we plug that value in to the given formula in place of σ.

(b)

```
R = 5000
calc_bootsrap_stddev <- function (sample) {
  bootstrap <- sample(Y.star,size=n,replace=T)
  mean(bootstrap)
}

bootsrap.means <- replicate(R, calc_bootsrap_stddev(20))
cat("std dev of the sample mean (bootstrap):",sd(bootsrap.means))
```

```
## std dev of the sample mean (bootstrap): 273901.3
```

(c)

```
n = 80
Y.star = sample(df$Salary,size=n,replace=T)
sample.sigma <- sd(Y.star)
cat("std dev of the sample mean (formulaic):",sample.sigma / sqrt(n),'\n')
```

```
## std dev of the sample mean (formulaic): 152665
```

```
bootsrap.means <- replicate(R, calc_bootsrap_stddev(20))
cat("variance of the sample mean (bootstrap):",sd(bootsrap.means))
```

```
## variance of the sample mean (bootstrap): 149267.1
```

The result from (a) and (b) are fairly similar. The above result improves when $n = 80$, as shown.

3. Binomial sampling

(a)

$$F_Y(y) = P(Y \leq y) = \sum_{i=0}^{\lfloor y \rfloor} \binom{7}{y} 0.1^i 0.9^{7-i}$$

(b)

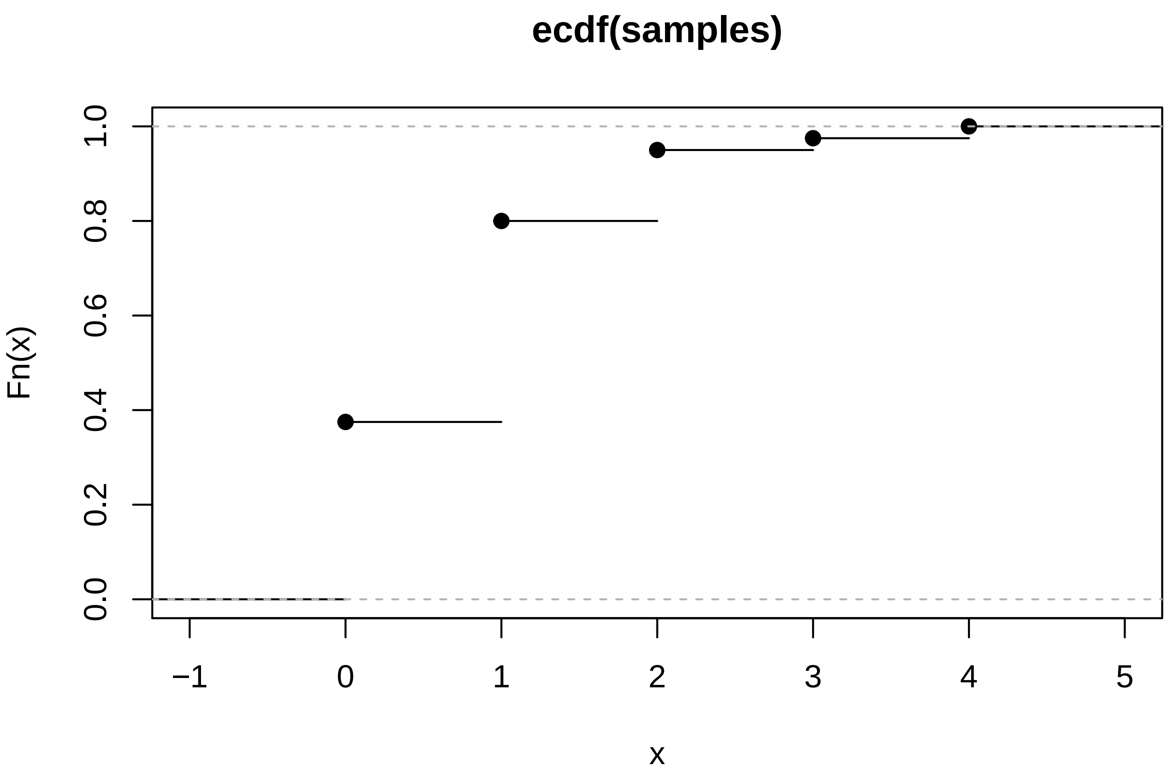
$$
\begin{aligned}
Var(\hat{F}_n(y)) &= Var(\frac{1}{n}\sum_{i=1}^{n} Var(\mathbf{1}_{Y_i \le y})) \\
&= \frac{1}{n^2}\sum_{i=1}^{n} Var(\mathbf{1}_{Y_i \le y}) \ (\because Y_i\text{'s are sampled iid}) \\
&= \frac{F_Y(y)(1 - F_Y(y))}{n}
\end{aligned}
$$

(c)

```r
num.trails <- 7
n <- 40
prob <- 0.1
samples <- rbinom(n, num.trails, prob)
```

i.

```r
plot(ecdf(samples))
```



ii.

It's pretty similar.

(d)

i.

```
ecdf.2s <- replicate(R, {
  samples <- rbinom(n, num.trails, prob)
  ecdf(samples)(2)
})
cat("standard deviation of F^n(2)",sd(ecdf.2s))
```

```
## standard deviation of F^n(2) 0.02554442
```

ii.

```
cdf.2 <- dbinom(0, num.trails, prob) +
+ dbinom(1, num.trails, prob) +
+ dbinom(2, num.trails, prob)

cat("standard deviation of sqrt of Var(F^n(2))",sqrt(cdf.2 * (1 - cdf.2) / n))
```

```
## standard deviation of sqrt of Var(F^n(2)) 0.02501572
```

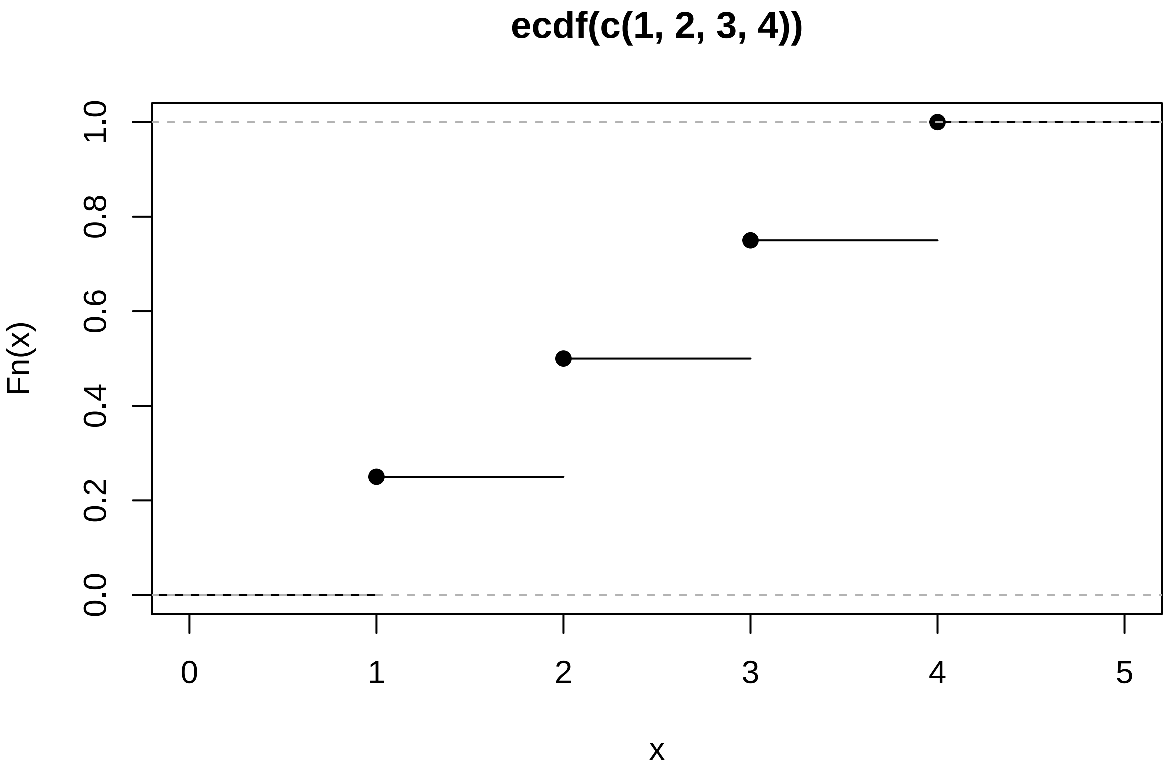
(e)

$Var(\hat{F}_n(2))$ is the variance across the empirical cumulative distribution function evaluated at 2. $S^2_{\hat{F}_n(2)}$ is the sample variance of the empirical cumulative distribution function evaluated at 2. Basically, the latter is the sample estimate of the former.

4. Quantiles

$n = 4$ Case

```
plot(ecdf(c(1,2,3,4)))
```



For $p = 0$, $Q_4(p) = min\{Y_1, Y_2, Y_3, Y_4\} = Y_{(1)} = 1$ by definition. $Y(\lceil 4p \rceil) = Y_{(0)}$, which does not exist. So the proposition doesn't hold for $p = 0$. But it will hold for cases when $p \neq 0$, as shown below.

For $0 < p \leq 0.25$,

$Q_4(p) = 1$. On the other hand, $Y(\lceil 4p \rceil) = Y_{(1)} = 1$

For $0.25 < p \leq 0.5$,

$Q_4(p) = 2$. On the other hand, $Y(\lceil 4p \rceil) = Y_{(2)} = 2$

For $0.5 < p \leq 0.75$,

$Q_4(p) = 3$. On the other hand, $Y(\lceil 4p \rceil) = Y_{(3)} = 3$

For $0.75 < p \leq 1$,

$Q_4(p) = 4$. On the other hand, $Y(\lceil 4p \rceil) = Y_{(4)} = 4$

General Case

Assume the samples are increasingly ordered. For any $p \in [0, 1]$, if np is not an integer, $\exists j \in 1, ..., n$ such that $\frac{j-1}{n} < p < \frac{j}{n}$. Hence, $Q_n(p) = Y_{(j)} = Y_{\lceil np \rceil}$.

if np is an integer, $p = \frac{j}{n}$. Hence, $Q_n(p) = inf\{y \in \mathbb{R} : \frac{j}{n} \leq \hat{F}_n(y)\} = Y(j) = Y(np) = Y_{\lceil np \rceil}$.

We have $Q_n(p) = Y_{\lceil np \rceil}$ in both cases.

5. Delta Method

(i)

Let $g(x) = logx$. Since g is continuously differentiable for $x > 0$, from delta method,

$$\sqrt{n}(g(\bar{Y}) - g(\mu)) \to N(0, \mu^2 g'(\mu)^2)$$

Hence,

$$\sqrt{n}(log\bar{Y} - log\mu) \to N(0, 1)$$

in distribution.

(ii)

Similarly to (i), let $g(x) = \sqrt{x}$. Since g is continuously differentiable for $x > 0$. Then,

$$\sqrt{n}(\sqrt{\bar{Y}} - \sqrt{\mu}) \to N(0, \frac{1}{4\mu})$$

in distribution.

(iii)

Similarly to (i), let $g(x) = logx$. Since g is continuously differentiable for $x \neq 0$. Then,

$$\sqrt{n}(\frac{1}{\bar{Y}} - \frac{1}{\mu}) \to N(0, \frac{1}{\mu^4})$$

in distribution.