# Lab 7

Kyle Mukire and 505832075

2022-05-22

## Contents



```
## Date last run: 2022-05-22
```

```
## Hello World!
```

# 1 Examples

Requires library xtable.

## 1.1 NWSL 2017-2021 Season Team-Game Data

```
## Read in our data
xdf <- read.table("NWSL_gameTeam.tsv", sep="\t", header=TRUE)

head(xdf, n=6)
```

```
##   season     date                     gameLeague                  team HT_VT Min Gls Ast
## 1   2017 20170415 National Women's Soccer League      Washington Spirit    HT 990   0   0
## 2   2017 20170415 National Women's Soccer League North Carolina Courage    VT 990   1   1
## 3   2017 20170415 National Women's Soccer League     Portland Thorns FC    HT 990   2   1
## 4   2017 20170415 National Women's Soccer League          Orlando Pride    VT 991   0   0
## 5   2017 20170415 National Women's Soccer League           Houston Dash    HT 990   2   2
## 6   2017 20170415 National Women's Soccer League      Chicago Red Stars    VT 990   0   0
```

```
##   PK PKatt CrdY CrdR
## 1  0     0    2    0
## 2  0     0    0    0
## 3  1     1    0    0
## 4  0     0    0    0
## 5  0     0    1    0
## 6  0     0    1    0
```

```
### table(xdf[ , "team"])
```

This data set was made by processing data obtained from FBREF.com

```
xbrks <- seq(-0.5, max(xdf[ , "Gls"])+0.5, by=1)
par(cex=0.65)
hist(xdf[ , "Gls"], breaks=xbrks, main="Team-Game Goals Scored, NWSL 2017-2021 Season")
```

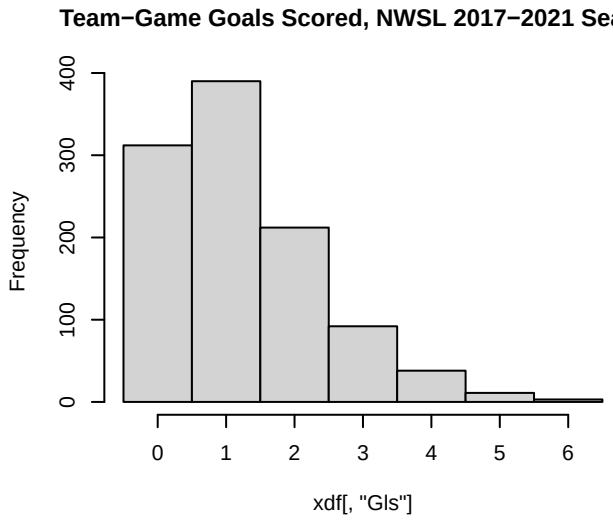



Figure 1: Distribution Team Goals scored by game.

Let's also create a frequency table.

```
library(xtable)
options(xtable.comment = FALSE)

xtbl <- table("goals"=xdf[ , "Gls"])
xtbl <- as.data.frame(xtbl)
print(xtable(xtbl, caption="NWSL game-team goals scored."), include.rownames=FALSE)
```

Calculate some summaries.

| goals | Freq |
|---|---|
| 0 | 312 |
| 1 | 390 |
| 2 | 212 |
| 3 | 92 |
| 4 | 38 |
| 5 | 11 |
| 6 | 3 |

Table 1: NWSL game-team goals scored.

```
xvarnames <- c("Gls", "Ast", "PK", "PKatt", "CrdY", "CrdR")

xmins <- apply(xdf[ , xvarnames], 2, min)

xmeans <- apply(xdf[ , xvarnames], 2, mean)

xmedians <- apply(xdf[ , xvarnames], 2, median)

xsds <- apply(xdf[ , xvarnames], 2, sd)

xIQRs <- apply(xdf[ , xvarnames], 2, IQR)

xmaxs <- apply(xdf[ , xvarnames], 2, max)

xsummaries <- rbind(xmins, xmeans, xmedians, xsds, xIQRs, xmaxs)

rownames(xsummaries) <- c("min", "mean", "median", "sd", "IQR", "max")

print(xtable(xsummaries, caption="NWSL game-team summary statistics."), include.rownames=TRUE)
```

| | Gls | Ast | PK | PKatt | CrdY | CrdR |
|---|---|---|---|---|---|---|
| min | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| mean | 1.24 | 0.85 | 0.08 | 0.12 | 1.05 | 0.03 |
| median | 1.00 | 1.00 | 0.00 | 0.00 | 1.00 | 0.00 |
| sd | 1.16 | 0.98 | 0.29 | 0.35 | 0.96 | 0.19 |
| IQR | 2.00 | 1.00 | 0.00 | 0.00 | 2.00 | 0.00 |
| max | 6.00 | 6.00 | 2.00 | 2.00 | 5.00 | 2.00 |

Table 2: NWSL game-team summary statistics.

Let's take a look at average game-team goals by season

```
xagg <- aggregate(xdf[ , "Gls"], by=list(xdf[ , "season"]), mean)

colnames(xagg) <- c("Season", "Goals")

print(xtable(xagg, caption="NWSL avg game-team goals scored by season."), include.rownames=FALSE)
```

| Season | Goals |
|---|---|
| 2017 | 1.39 |
| 2018 | 1.27 |
| 2019 | 1.26 |
| 2020 | 1.18 |
| 2021 | 1.10 |

Table 3: NWSL avg game-team goals scored by season.

```
#x <- as.factor(xagg[ , "Season"])
x <- xagg[ , "Season"]
y <- xagg[ , "Goals"]
par(cex=0.65)
plot(x, y, type="l", col="#3355BB", main="Avg Team-Game Goals Scored by NWSL Season",
     xaxt="n", xlab="Season", ylab="Avg Team-Game Goals", lwd=2)
axis(1, c(2017, 2018, 2019, 2020, 2021), c("2017", "2018", "2019", "2020", "2021"))
```

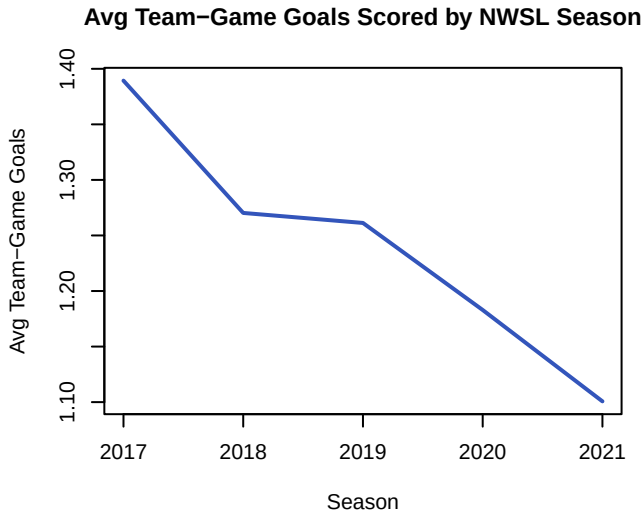



Figure 2: NWSL Avg Team-Game Goals by Season.

### 1.1.1 Simple Random Sampling

This is an illustration.

We are going to draw 200,000 random samples of size $n = 16$, with replacement, from our NWSL team-game data, and for each simulation calculate the sample average. We'll then use a histogram to graphically convey this empirical sampling distribution.

By the way, when we sample with replacement we are defining our population as being comprised of an infinite collection of copies of our data.

We are simulating unbiased sampling — in particular, simple random sampling.

```r
set.seed(777)

nn <- 200000

N <- 16

xavg_vec <- numeric(nn)

for(i in 1:length(xavg_vec)) {
    xavg_vec[i] <- mean( sample(xdf[ , "Gls"], size=N, replace=TRUE) )
}
```

```r
par(cex=0.65)
hist(xavg_vec, main="Empiric Sampling Dist. NWSL Team-Game Goals, n=16")
abline(v=mean(xavg_vec), col="#33BB33AA", lwd=7)
abline(v=mean(xdf[ , "Gls"]), col="#BB3333AA", lwd=3)
```

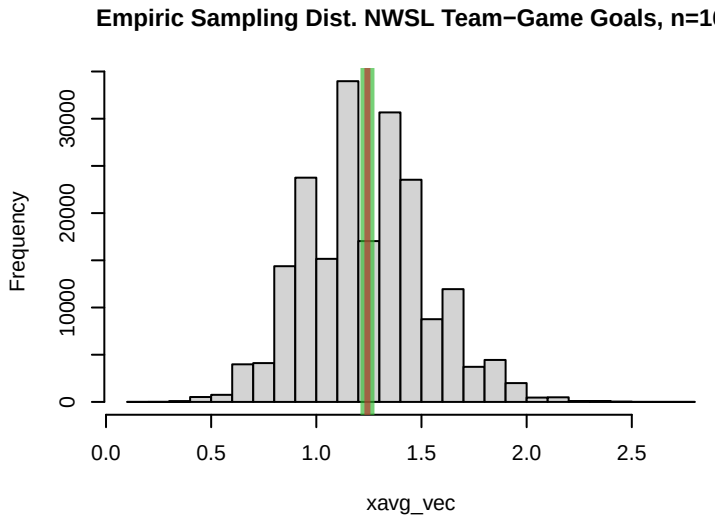



Figure 3: Empirical Sampling Distribution of mean Team-Game goals from NWSL, sample size of 16. The red line marks the true population parameter mean team-game goals; the green, the mean of all the simulated sample means.

The fact that the red and green lines in Figure 3 closely mark the same value tell us that the population mean and the average of the sample means closely agree. The suggestion is that the method of sampling (the sample() function in **R**), is unbiased.

### 1.1.2 Deliberately Biased Sampling

We can deliberately illustrate bias by explicitly not drawing samples with impartiality. For example, we can draw team-game goals only from the 2017 season.

```r
set.seed(777)

nn <- 200000

N <- 16

xavg_vec <- numeric(nn)

xbiasmask <- xdf[ , "season"] %in% "2017"
#xbiasmask <- xdf[ , "team"] %in% "Washington Spirit"

for(i in 1:length(xavg_vec)) {
    xavg_vec[i] <- mean( sample(xdf[ xbiasmask, "Gls"], size=N, replace=TRUE) )
}
```

```r
par(cex=0.65)
hist(xavg_vec, main="Biased Empiric Sampling Dist. NWSL Team-Game Goals, n=16")
abline(v=mean(xavg_vec), col="#33BB33AA", lwd=7)
abline(v=mean(xdf[ , "Gls"]), col="#BB3333AA", lwd=3)
```

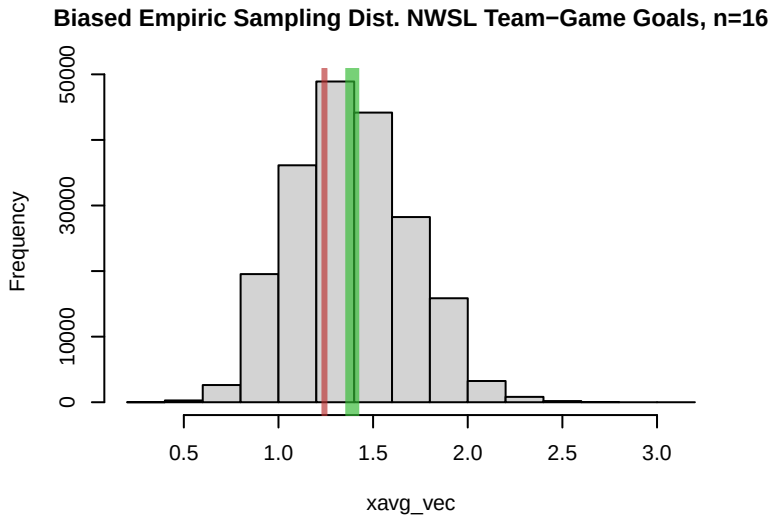



Figure 4: Empirical Sampling Distribution of mean Team-Game goals from NWSL, sample size of 16, where sample was drawn only from 2017 season. The red line marks the true population parameter mean team-game goals; the green, the mean of all the simulated sample means.

The fact that the red and green lines in Figure 4 do not mark the same value tell us that the population mean and the average of the sample means are different. The suggestion is that the method of sampling (only drawing from one season) is **biased**.

## 1.2 Confidence Interval for Population Proportion

### 1.2.1 Super Bowl Coin Tosses

```
## Read in our data
sbdf <- read.table("SuperBowl_coinTosses.tsv", sep="\t", header=TRUE)

head(sbdf, n=6)
```

```
##   SuperBowl             Matchup CoinToss Coin.TossWinner GameWinner
## 1         1   Chiefs vs Packers    Heads         Packers    Packers
## 2         2  Packers vs Raiders    Tails         Raiders    Packers
## 3         3       Colts vs Jets    Heads            Jets       Jets
## 4         4   Vikings vs Chiefs    Tails         Vikings     Chiefs
## 5         5    Colts vs Cowboys    Tails         Cowboys      Colts
## 6         6 Cowboys vs Dolphins    Heads        Dolphins    Cowboys
##   CoinTossWinnerEqualGameWinner.
## 1                            Yes
## 2                             No
## 3                            Yes
## 4                             No
## 5                             No
## 6                             No
```

This data set was obtained from https://www.sportsbettingdime.com/guides/resources/super-bowl-coin-toss-history/

Imagine that these 56 observations were randomly realized from an infinite population of Super Bowl coin tosses.

Let's create a 68% Confidence Interval for the true, unknown population parameter proportion of Heads.

```
n <- nrow(sbdf)

xheads <- as.integer(sbdf[, "CoinToss"] %in% "Heads")

phat <- mean(xheads)
phat
```

```
## [1] 0.4821429
```

```
### OR

phat <- sum(xheads) / length(xheads)
phat
```

```
## [1] 0.4821429
```

```r
var_phat <- phat * (1 - phat)
var_phat
```

```
## [1] 0.2496811
```

```r
est_SE <- sqrt(var_phat / n)
est_SE
```

```
## [1] 0.06677269
```

We need to calculate our estimated margin of error from our confidence level.

We're using the normal approximation. We know from the empiric rule that $-1 < Z < 1$ chops out about the middle 68% of the normal distribution.

We can also have **R** do this for us.

```r
z_low <- qnorm( (1 - 0.68) / 2 )
z_low
```

```
## [1] -0.9944579
```

```r
z_high <- qnorm( 0.68 + (1 - 0.68) / 2 )
z_high
```

```
## [1] 0.9944579
```

Let's just use $z_L = -1$ and $z_H = 1$

```r
z_low <- -1
z_high <- 1

CI_low <- phat + z_low * est_SE

CI_high <- phat + z_high * est_SE
```

We are 68% confident that the true proportion of heads within this imaginary, infinite population of Super Bowl coin tosses is between 0.4154 and 0.5489.

## 2 Your Work

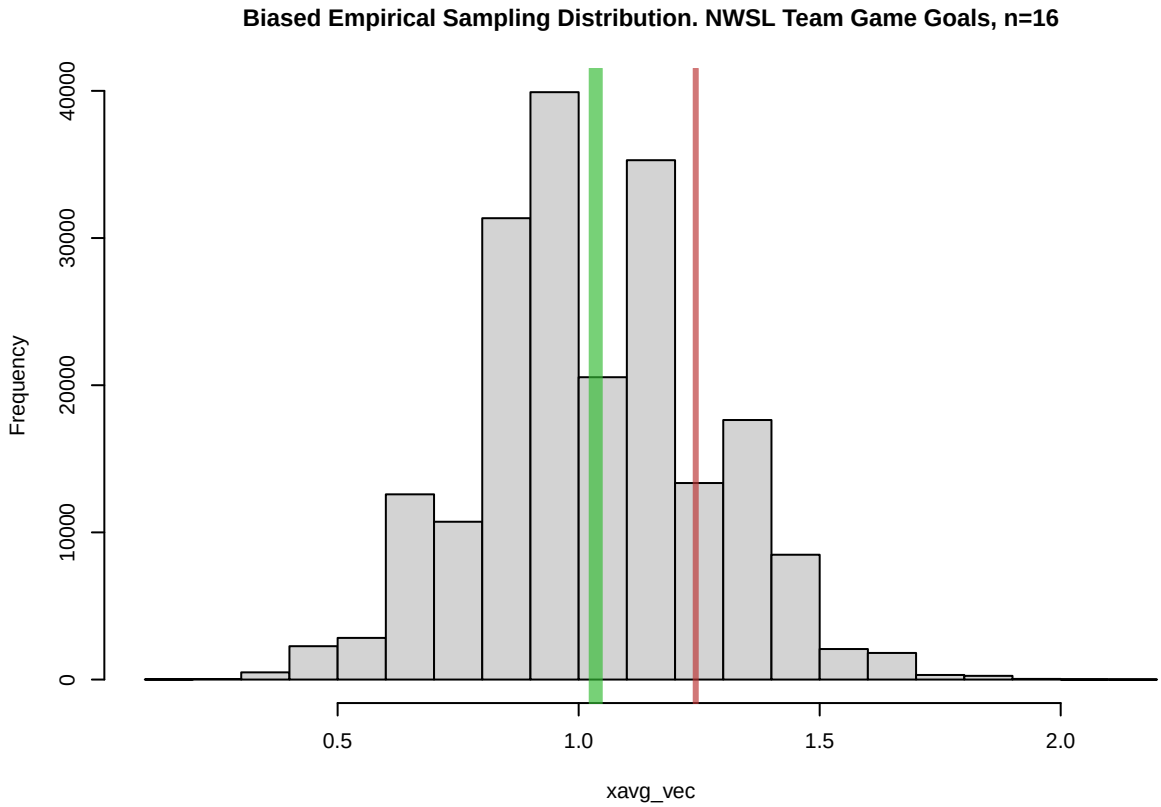
Make sure to edit the "author" information in the YAML header near the top to include your name and UID.

Complete/answer the following.

1 — Consider Figure 4. Repeat the process we used to illustrate biased sampling, except instead of drawing only from one season, draw your sample from all seasons but only rows where the team is "Washington Spirit". Comment on your findings.

```
set.seed(777)
nn <- 200000
N <- 16
xavg_vec <- numeric(nn)
# xbiasmask <- xdf[ , "season"] %in% "2017"
xbiasmask <- xdf[ , "team"] %in% "Washington Spirit"
for(i in 1:length(xavg_vec)) {
xavg_vec[i] <- mean(sample(xdf[ xbiasmask, "Gls"], size=N, replace=TRUE) )
}
```

```
par(cex=0.65)
hist(xavg_vec, main="Biased Empirical Sampling Distribution. NWSL Team Game Goals, n=16")
abline(v=mean(xavg_vec), col="#33BB33AA", lwd=7)
abline(v=mean(xdf[ , "Gls"]), col="#BB3333AA", lwd=3)
```




Through examining this we can see that the sampling is biased. The gree and red line are not intersecting.

Green represents the Washington Spirit while red represents all the teams. Since the green line is lower than the red line that shows that the Washington Spirit performes below average.

2 — Show that we have met the requirements for using the normal approximation to the binomial model in our CI calculation above.

1. When the probability of success if near 0 or 1
2. n$p$ *and* $n$(1-p) >=5 or 10 -This ensures probability of success is large enough or has enough trials for a probability of success.

```
n2 <- nrow(sbdf)

n* phat
```

```
## [1] 27
```

```
n* (1-phat)
```

```
## [1] 29
```

As phat is from the binomial distribution, and both of these are greater than 10, meeting the requirements above.

3 — Imagine our NWSL data set is actually a randomly realized collection of games from an infinite collection of games that could have resulted. Thinking about the population parameter proportion of the occurrence of a team drawing one or more Red Cards in a game, create a 90% CI, a 95% CI, and a 99% CI. Comment and interpret your results. Also show that we have met the requirements for using the normal approximation to the binomial model.

```
### here's a head start

xteamGame_redCard <- as.integer( xdf[ , "CrdR" ] > 0 )

phat_rc <- mean(xteamGame_redCard)
phat_rc
```

```
## [1] 0.03213611
```

```
est_SE_rc <- sqrt( phat_rc * (1 - phat_rc) / length(xteamGame_redCard) )
est_SE_rc
```

```
## [1] 0.005422018
```

```
z_low_90 <- qnorm( (1 - 0.9) / 2 )
z_high_90 <- qnorm( 0.9 + ( 1 - 0.9 ) / 2 )

z_low_95 <- qnorm( (1 - 0.95) / 2 )
z_high_95 <- qnorm( 0.95 + ( 1 - 0.95 ) / 2 )

z_low_99 <- qnorm( (1 - 0.99) / 2 )
z_high_99 <- qnorm( 0.99 + (1 - 0.99 ) / 2 )

c(phat_rc + est_SE_rc * z_low_90, phat_rc + est_SE_rc * z_high_90)
```

```
## [1] 0.02321768 0.04105453
```

```
c(phat_rc + est_SE_rc * z_low_90, phat_rc + est_SE_rc * z_high_95)
```

```
## [1] 0.02321768 0.04276307
```

```
c(phat_rc + est_SE_rc * z_low_90, phat_rc + est_SE_rc * z_high_99)
```

```
## [1] 0.02321768 0.04610230
```

```
#90% sure
#95% sure
#99% sure
```

Smaller intervals create less confidence because they hold fewer values. This causes the interval size to increase as well as confidence increase because the expected value is in the given interval. It's a compromise between precision and confidence.

```
n2 <- nrow(xdf)

n2 * phat_rc
```

```
## [1] 34
```

```
n2* (1-phat_rc)
```

```
## [1] 1024
```

As phat_rc comes from the binomial distribution and with both values being greater than 10 which means they meet the standards of the normal approx model.