Lab 9

Apurva Shah

2022-06-05

Contents


```
## Date last run: 2022-06-05
```

```
## Hello World!
```

Requires library xtable.

Examples

Heavy Skewed Parent Distribution

Let's start with an extreme case.

Image a lottery at a fair or event. It costs $2 to purchase a ticket. There's a 1-in-20 chance of winning $10 (that's an $8 net), and a 1-in-10,000 chance of winning $1000 (that's a $998 net).

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

library(xtable)

xdomain <- c(-2, 8, 998)

p_big <- 1 / 10^4
p_small <- 1 / 20
p_lose <- 1 - p_big - p_small

xcprobs <- paste0(10000 * c(p_lose, p_small, p_big), "/", 10000)
xcprobs
```

```
## [1] "9499/10000" "500/10000"  "1/10000"
```

```r
df_ptble <- data.frame("NetWin"=xdomain, "Probability"=xcprobs)
```

```r
print(xtable(df_ptble, caption="Probability Table for Lottery net winnings."),
      include.rownames=FALSE)
```

NetWin	Probability
-2.00	9499/10000
8.00	500/10000
998.00	1/10000

Table 1: Probability Table for Lottery net winnings.

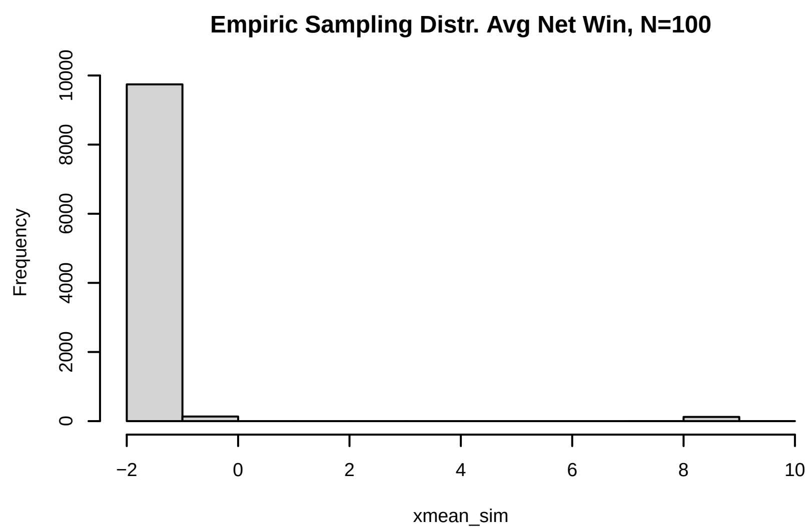
What is the sampling distribution of average net winnings for 100 tickets?

```r
nn <- 10000 ### number of simulations

N <- 100 ### sample size

xmean_sim <- numeric(nn)

for(ii in 1:nn) {
    x_sim_win <- sample(xdomain, size=N, prob=c(p_lose, p_small, p_big), replace=TRUE)
    xmean_sim[ii] <- mean(x_sim_win)
}
```

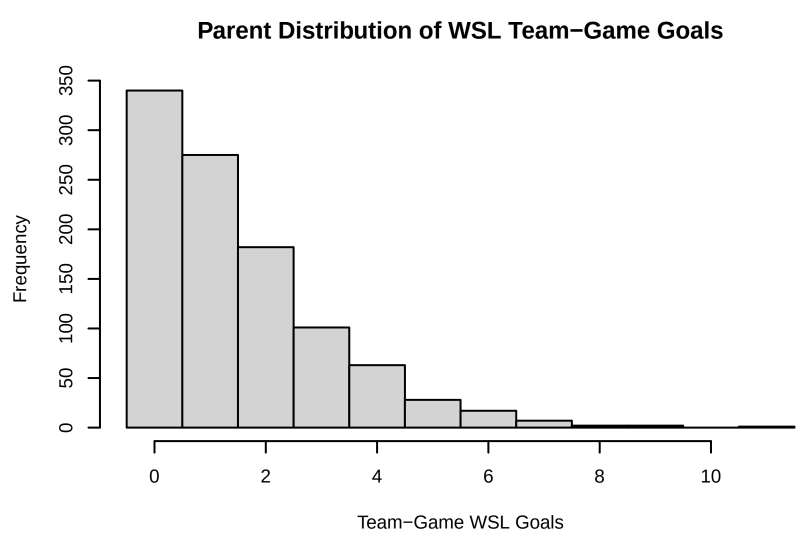


Nowhere close to bell-shaped.

Moderately Skewed Parent Distribution

Women's Super League Team-Game Goals

```
xdf <- read.table( "WomensSuperLeague_teamDate.tsv", header=TRUE, sep="\t" )
```



Let's imagine our population is an infinite collection of games just like these we have (i.e., sample with replacement) and simulate the sampling distribution of the mean Team-Game Goals for 4 different sample sizes.

```
nn <- 40000

xmean_a <- numeric(nn)
xmean_b <- numeric(nn)
xmean_c <- numeric(nn)
xmean_d <- numeric(nn)

for(ii in 1:nn) {
    xmean_a[ii] <- mean(sample(xdf[ , "Gls"], size=4, replace=TRUE))
    xmean_b[ii] <- mean(sample(xdf[ , "Gls"], size=16, replace=TRUE))
    xmean_c[ii] <- mean(sample(xdf[ , "Gls"], size=64, replace=TRUE))
    xmean_d[ii] <- mean(sample(xdf[ , "Gls"], size=100, replace=TRUE))
}
```

Looking at Figure @ref(fig:goalsESD), we see the simulated empiric sampling distribution when the sample size is 9 is right skewed. When 16, the right skewness is reduced, but still visually present. For sample size 100, the sampling distribution looks fairly symmetrical.

Let's repeat the process for the sampling distribution of our T-statistic.

```
nn <- 40000

xmeanT_a <- numeric(nn)
xmeanT_b <- numeric(nn)
xmeanT_c <- numeric(nn)
xmeanT_d <- numeric(nn)

n_a <- 4
n_b <- 16
n_c <- 64
n_d <- 100

xmu <- mean(xdf[ , "Gls"])

for(ii in 1:nn) {

    xa <- sample(xdf[ , "Gls"], size=n_a, replace=TRUE)
    xmeanT_a[ii] <- (mean(xa) - xmu) / sqrt( var(xa) / n_a )

    xb <- sample(xdf[ , "Gls"], size=n_b, replace=TRUE)
    xmeanT_b[ii] <- (mean(xb) - xmu) / sqrt( var(xb) / n_b )

    xc <- sample(xdf[ , "Gls"], size=n_c, replace=TRUE)
    xmeanT_c[ii] <- (mean(xc) - xmu) / sqrt( var(xc) / n_c )

    xd <- sample(xdf[ , "Gls"], size=n_d, replace=TRUE)
    xmeanT_d[ii] <- (mean(xd) - xmu) / sqrt( var(xd) / n_d )
```

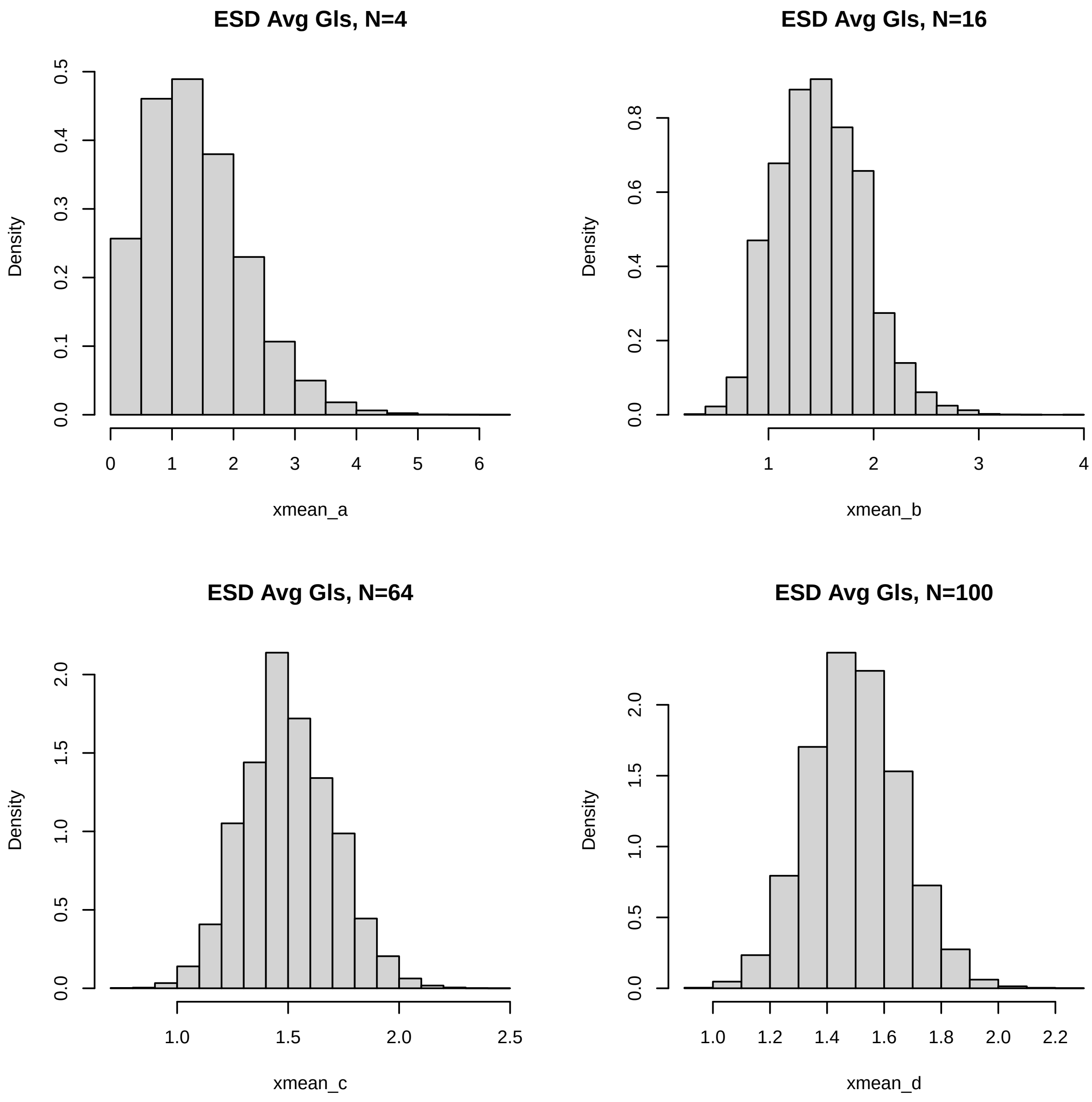


Figure 1: Empiric Sampling Distributions of average WSL Team-Game Goals for 4 different sample sizes.

```r
}

tdom <- seq(-5, 5, length=300)

tden_a <- dt(tdom, n_a-1)
tden_b <- dt(tdom, n_b-1)
tden_c <- dt(tdom, n_c-1)
tden_d <- dt(tdom, n_d-1)
```

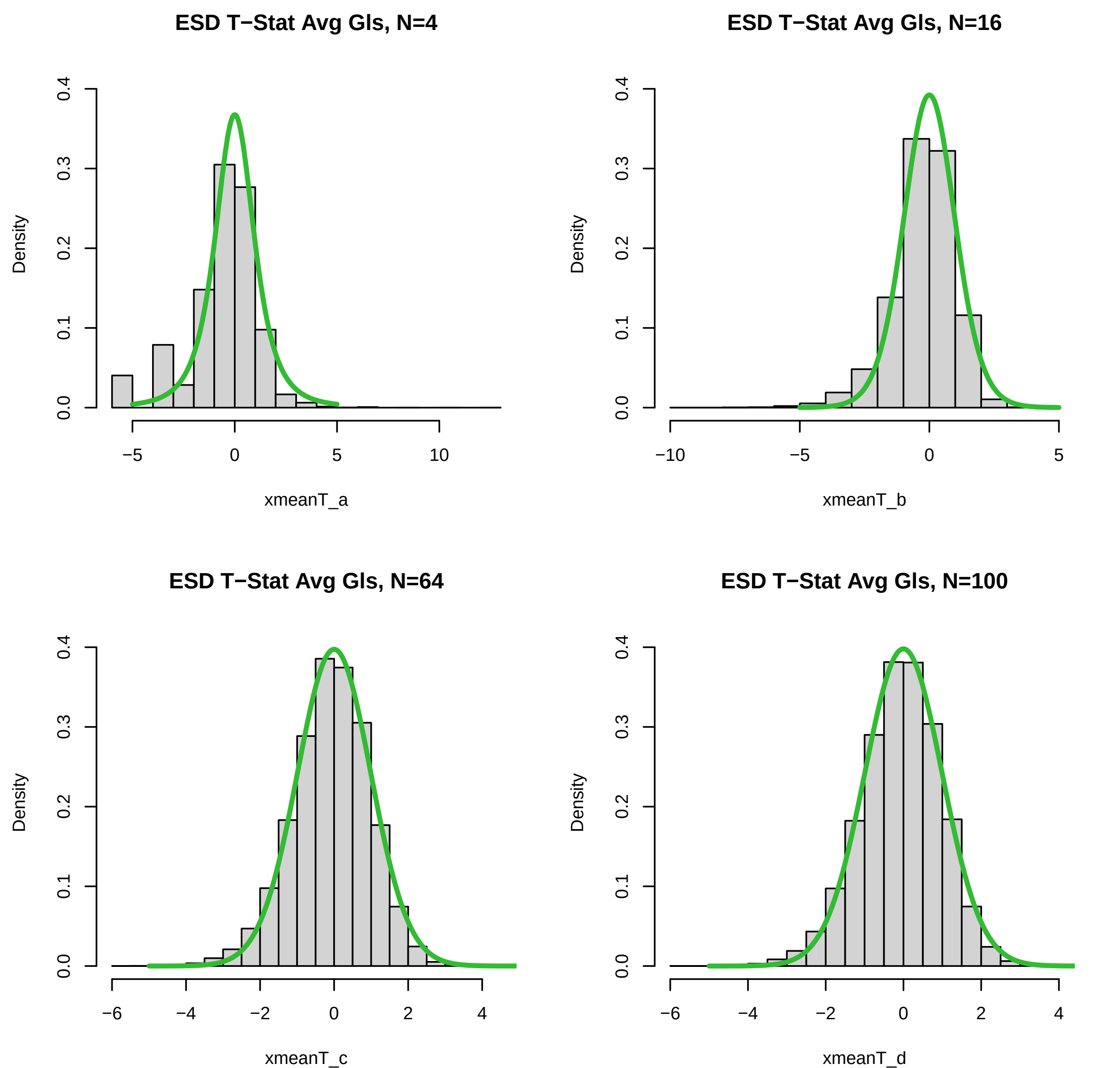


Figure 2: Empiric Sampling Distributions of T-Statistic for average WSL Team-Game Goals for 4 different sample sizes. Green path shows T-Distribution for N-1 degrees of freedom.

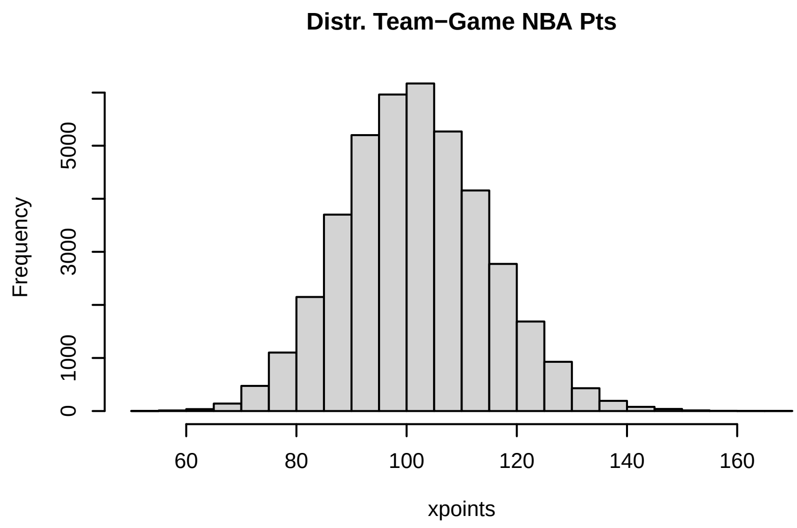
Looking at Figure @ref(fig:goalsTstatESD), we see the simulated empiric sampling distributions are poorly approximated by the respective T-Distribution model, except perhaps where the sample size is 100.

Close to Bell-Shaped Parent Distribution

NBA Points

```
df_nba <- read.table(file.path("NBA_teamGame.tsv"), sep="\t", header=TRUE)
tail(df_nba)
```

```
##           date matchup team min tpm tpa oreb dreb ast stl blk to pf pts HA
## 40515 20210516 LAC@OKC  LAC 240  10  43   16   28  17   8   3  3 14 112 -1
## 40516 20210516 LAC@OKC  OKC 240   8  26   14   40  20   1  12 15 11 117  1
## 40517 20210516 DEN@POR  DEN 240  14  37   10   26  20   8   2  6 20 116 -1
## 40518 20210516 DEN@POR  POR 240  18  43   11   40  24   3   6 13 16 132  1
## 40519 20210516 UTZ@SAC  UTZ 240  18  39    9   41  28   9   3 16 14 121 -1
## 40520 20210516 UTZ@SAC  SAC 240   9  30    5   34  24  10   5 12 22  99  1
```




```
nn <- 40000

xmeanT_a <- numeric(nn)
xmeanT_b <- numeric(nn)
xmeanT_c <- numeric(nn)
xmeanT_d <- numeric(nn)

n_a <- 9
n_b <- 16
n_c <- 64
n_d <- 100

xmu <- mean(xpoints)
```

```r
for(ii in 1:nn) {

    xa <- sample(xpoints, size=n_a, replace=FALSE)
    xmeanT_a[ii] <- (mean(xa) - xmu) / sqrt( var(xa) / n_a )

    xb <- sample(xpoints, size=n_b, replace=FALSE)
    xmeanT_b[ii] <- (mean(xb) - xmu) / sqrt( var(xb) / n_b )

    xc <- sample(xpoints, size=n_c, replace=FALSE)
    xmeanT_c[ii] <- (mean(xc) - xmu) / sqrt( var(xc) / n_c )

    xd <- sample(xpoints, size=n_d, replace=FALSE)
    xmeanT_d[ii] <- (mean(xd) - xmu) / sqrt( var(xd) / n_d )

}

tdom <- seq(-5, 5, length=300)

tden_a <- dt(tdom, n_a-1)
tden_b <- dt(tdom, n_b-1)
tden_c <- dt(tdom, n_c-1)
tden_d <- dt(tdom, n_d-1)
```

Looking at Figure @ref(fig:pointsTstatESD), we see that even when the sample size is only 9, it looks like the T-Distribution model closely approximates our simulated empiric sampling distribution. Recall our parent distribution appears to be close to normal.

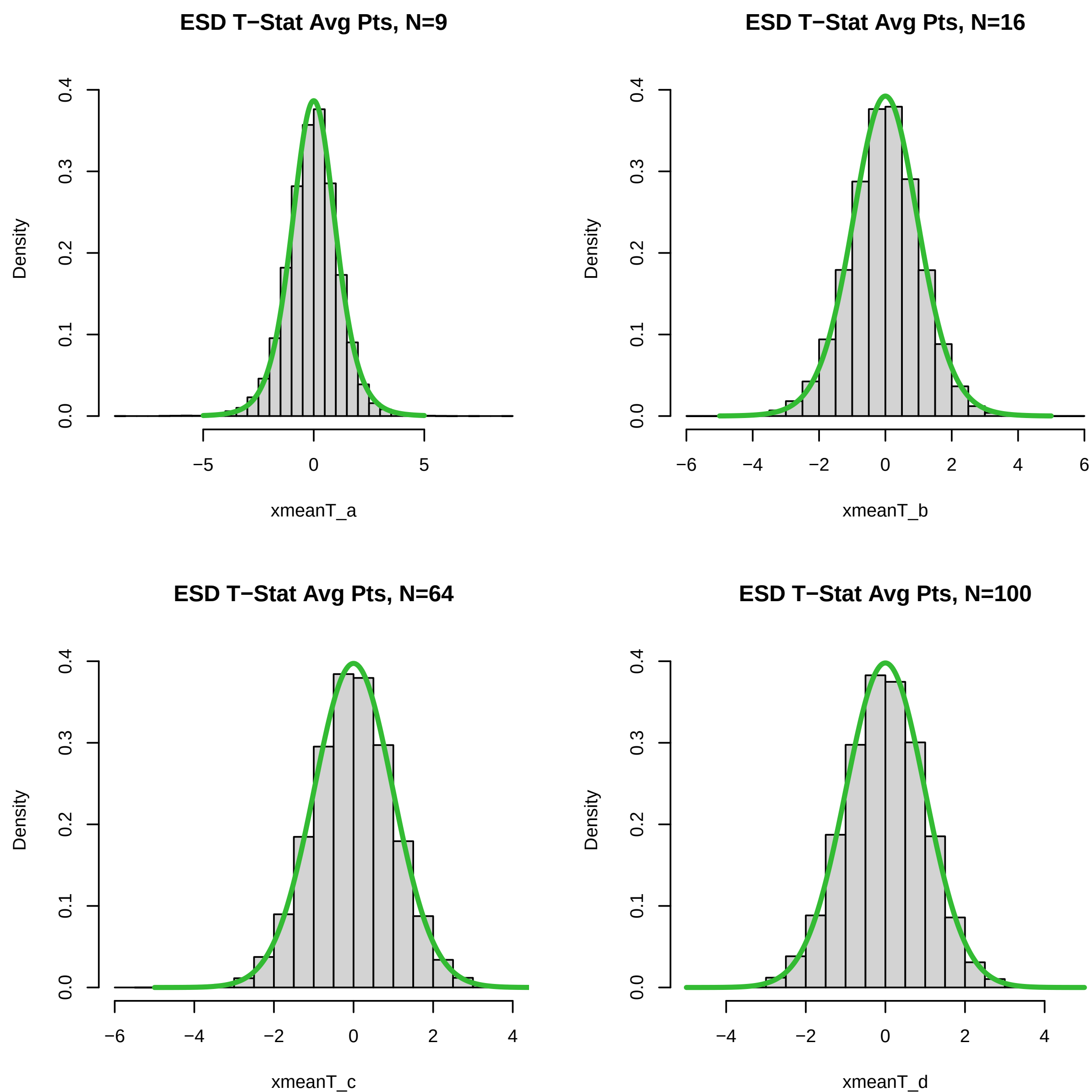


Figure 3: Empiric Sampling Distributions of T-Statistic for average NBA Team-Game Points for 4 different sample sizes. Green path shows T-Distribution for N-1 degrees of freedom.

Hypothesis Test of Population Mean

As if we haven't had enough fun already with all these simulations, let's get down to illustrating hypothesis testing of a population mean in **R**.

We should start with make-believe. Let's make-believe we don't have access to the full NBA Team-Game data we just looked at.

Suppose someone claims that the average points scored by teams in an NBA game from the 2004-2005 season through the 2020-2021 season is greater than 100 points.

$H_0 : \mu = 100$

$H_a : \mu > 100$

Let's set $\alpha = 0.01$

Here's our sample (randomly obtained):

```
df_nba_sample <- read.table(file.path("NBA_teamGame_sample.tsv"), sep="\t", header=TRUE)
head(df_nba_sample)
```

```
##       date matchup team min tpm tpa oreb dreb ast stl blk to pf pts HA
## 1 20171209 PHI@CLE  PHI 240  11  33    7   32  31  10   2 19 21  98 -1
## 2 20041219 ORL@MIA  MIA 240   8  17    7   27  24   5   1  9 20 117  1
## 3 20090203 CHI@HOU  HOU 240   6  20   12   34  19   6   9 11 20 107  1
## 4 20110211 NOP@ORL  ORL 240   5  21    7   34  21   8   6 16 21  93  1
## 5 20210203 WAS@MIA  MIA 240  12  35    9   35  24  11   2 13 21 100  1
## 6 20070312 HOU@PHX  PHX 240   9  15   10   40  18   3   6 10 13 103  1
```

```
N <- nrow(df_nba_sample)
N
```

```
## [1] 300
```

```
x_bar <- mean(df_nba_sample[ , "pts"])

SE_est <- sqrt( var(df_nba_sample[ , "pts"]) / N )

t_stat <- (mean(df_nba_sample[ , "pts"]) - 100) / SE_est

#### right tail
pval <- 1 - pt(t_stat, N-1)
```

Our sample size is 300.

Our critical region is $[2.33888, \infty)$.

Our sample average is $\bar{x} = 102.28667$.

Our estimated standard error is $\mathsf{SE}[\bar{x}] = 0.75386$.

Our actual observed T test statistic is $t_{299} = 3.03327$.

Our p-value is 0.00132.

We conclude in favor of the alternative hypothesis, that the true average team-game points is greater than 100.

Confidence Interval for Population Mean

```
#### 95 %

t_low <- qt(0.025, N-1)

t_high <- qt(0.975, N-1)


CI_low <- x_bar + SE_est * t_low

CI_high <- x_bar + SE_est * t_high
```

We are 95% confident that the true average team-game points resides in $(100.803, 103.77)$

```
#### 99 %

t_low <- qt(0.005, N-1)

t_high <- qt(0.995, N-1)

CI_low <- x_bar + SE_est * t_low

CI_high <- x_bar + SE_est * t_high
```

We are 99% confident that the true average team-game points resides in $(100.332, 104.241)$

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 our hypothesis test and confidence intervals in the Examples Section above. Do you believe the T-distribution model accurately characterizes our T-stat estimator? Explain.

I think the T distribution model does accurately characterize our tstat estimator.

2 — Pretend we don't have access to the full NBA Team-Game data. Use the sample NBA data to test the claim, at $\alpha = 0.001$, that the true population average of 3-point shots made by teams per game is greater than 6. Also construct a 99% confidence interval. Interpret your findings. Do you believe the T-distribution model accurately characterizes our T-stat estimator? Explain.

```
library(tidyverse)
library(xtable)

# 2 --- Pretend we don't have access to the full NBA Team-Game data.  Use the sample NBA data to test t
# $\alpha=0.001$, that the true population average of 3-point shots made by teams per game is greater t
# Interpret your findings.  Do you believe the T-distribution model accurately characterizes our T-stat


# H0 :u=6
# Ha :u>6
# Let's set a = 0.001


xdf <- read.table("/Users/apurvashah/Documents/GitHub/stats10/lab10/studentKit/yourLab/NBA_teamGame_samp
head(xdf)
```

```
##       date matchup team min tpm tpa oreb dreb ast stl blk to pf pts HA
## 1 20171209 PHI@CLE  PHI 240  11  33    7   32  31  10   2 19 21  98 -1
## 2 20041219 ORL@MIA  MIA 240   8  17    7   27  24   5   1  9 20 117  1
## 3 20090203 CHI@HOU  HOU 240   6  20   12   34  19   6   9 11 20 107  1
## 4 20110211 NOP@ORL  ORL 240   5  21    7   34  21   8   6 16 21  93  1
## 5 20210203 WAS@MIA  MIA 240  12  35    9   35  24  11   2 13 21 100  1
## 6 20070312 HOU@PHX  PHX 240   9  15   10   40  18   3   6 10 13 103  1
```

```
N <- nrow(xdf)
x_bar <- mean(xdf[ , "tpm"])
SE_est <- sqrt( var(xdf[ , "tpm"]) / N )
t_stat <- (mean(xdf[ , "tpm"]) - 6) / SE_est
pval <- 1 - pt(t_stat, N-1)

# // We conclude in the favor of the null hypothesis, that the TPM made per game is not greater than si

t_low <- qt(0.005, N-1)
t_high <- qt(0.995, N-1)
CI_low <- x_bar + SE_est * t_low
CI_high <- x_bar + SE_est * t_high

paste("We are 99% confident that the true average three point shots made resides in", CI_low, CI_high, s
```

```
## [1] "We are 99% confident that the true average three point shots made resides in 7.26033748713594 8
```

```
nn <- 40000

xmean_a <- numeric(nn)
xmean_b <- numeric(nn)
xmean_c <- numeric(nn)
xmean_d <- numeric(nn)
for(ii in 1:nn) {
  xmean_a[ii] <- mean(sample(xdf[ , "tpm"], size=4, replace=TRUE))
  xmean_b[ii] <- mean(sample(xdf[ , "tpm"], size=16, replace=TRUE))
  xmean_c[ii] <- mean(sample(xdf[ , "tpm"], size=64, replace=TRUE))
  xmean_d[ii] <- mean(sample(xdf[ , "tpm"], size=100, replace=TRUE))
}


xmeanT_a <- numeric(nn)
xmeanT_b <- numeric(nn)
xmeanT_c <- numeric(nn)
xmeanT_d <- numeric(nn)

n_a <- 4
n_b <- 16
n_c <- 64
n_d <- 100
xmu <- mean(xdf[ , "tpm"])
for(ii in 1:nn) {
  xa <- sample(xdf[ , "tpm"], size=n_a, replace=TRUE)
  xmeanT_a[ii] <- (mean(xa) - xmu) / sqrt( var(xa) / n_a )
  xb <- sample(xdf[ , "tpm"], size=n_b, replace=TRUE)
  xmeanT_b[ii] <- (mean(xb) - xmu) / sqrt( var(xb) / n_b )
  xc <- sample(xdf[ , "tpm"], size=n_c, replace=TRUE)
  xmeanT_c[ii] <- (mean(xc) - xmu) / sqrt( var(xc) / n_c )
  xd <- sample(xdf[ , "tpm"], size=n_d, replace=TRUE)
  xmeanT_d[ii] <- (mean(xd) - xmu) / sqrt( var(xd) / n_d )
}
tdom <- seq(-5, 5, length=300)

tden_a <- dt(tdom, n_a-1)
tden_b <- dt(tdom, n_b-1)
tden_c <- dt(tdom, n_c-1)
tden_d <- dt(tdom, n_d-1)
```

3 — Consider our Lottery scenario. How large of a sample size do we need for the sampling distribution of average net winnings to start to look bell-shaped? Note that your computer may run for a long time.

```
xdf <- read.table("/Users/apurvashah/Documents/GitHub/stats10/lab10/studentKit/yourLab/NBA_teamGame_samp
head(xdf)
```

```
##       date matchup team min tpm tpa oreb dreb ast stl blk to pf pts HA
## 1 20171209 PHI@CLE  PHI 240  11  33    7   32  31  10   2 19 21  98 -1
## 2 20041219 ORL@MIA  MIA 240   8  17    7   27  24   5   1  9 20 117  1
## 3 20090203 CHI@HOU  HOU 240   6  20   12   34  19   6   9 11 20 107  1
## 4 20110211 NOP@ORL  ORL 240   5  21    7   34  21   8   6 16 21  93  1
## 5 20210203 WAS@MIA  MIA 240  12  35    9   35  24  11   2 13 21 100  1
## 6 20070312 HOU@PHX  PHX 240   9  15   10   40  18   3   6 10 13 103  1
```

```r
options(xtable.comment = FALSE)
table(xdf[, "HA"])
```

```
##
##  -1   1
## 143 157
```

```r
total<-143+157

xdomain <- c(-1, 1)
p_big <- 1 / 2
p_small <- 1 / 2
p_lose <- 1/2
p_win <- 157/total
p_lose <- 143/total

xcprobs <- paste0(10000 * c(p_win, p_lose), "/", 10000)
xcprobs
```

```
## [1] "5233.33333333333/10000" "4766.66666666667/10000"
```

```r
nn <- 100000 ### number of simulations
N <- 100 ### sample size
xmean_sim <- numeric(nn)
for(ii in 1:nn) {
  x_sim_win <- sample(xdomain, size=N, prob=c(p_lose, p_win), replace=TRUE)
  xmean_sim[ii] <- mean(x_sim_win)
}


## I set the sample size to 1,000,000 and then it looked like a bell curve for me.
```