

Stat 3202 Lab 12

Taylor Keckley

4/21/2020

Bootstrapping

Revisit the APGAR data from Lecture 16A: Usual method: c(8, 7, 6, 2, 5, 8, 7, 3) New method: c(9, 9, 7, 8, 10, 8, 6)

Part a)

Group all scores into one single sample of size 15. Create 95% bootstrap confidence intervals for the mean and standard deviation of the APGAR score.

```
apgar1 <- c(8, 7, 6, 2, 5, 8, 7, 3)
apgar2 <- c(9, 9, 7, 8, 10, 8, 6)
apgar <- c(apgar1, apgar2)
B <- 5000
xbar_storage <- rep(NA, B)
s_storage <- rep(NA, B)
n <- length(apgar)

for (i in 1:B) {
  apgar_boot <- sample(apgar, size=n, replace=TRUE)
  xbar_storage[i] <- mean(apgar_boot)
  s_storage[i] <- sd(apgar_boot)
}

quantile(xbar_storage, c(0.025, 0.975))
```

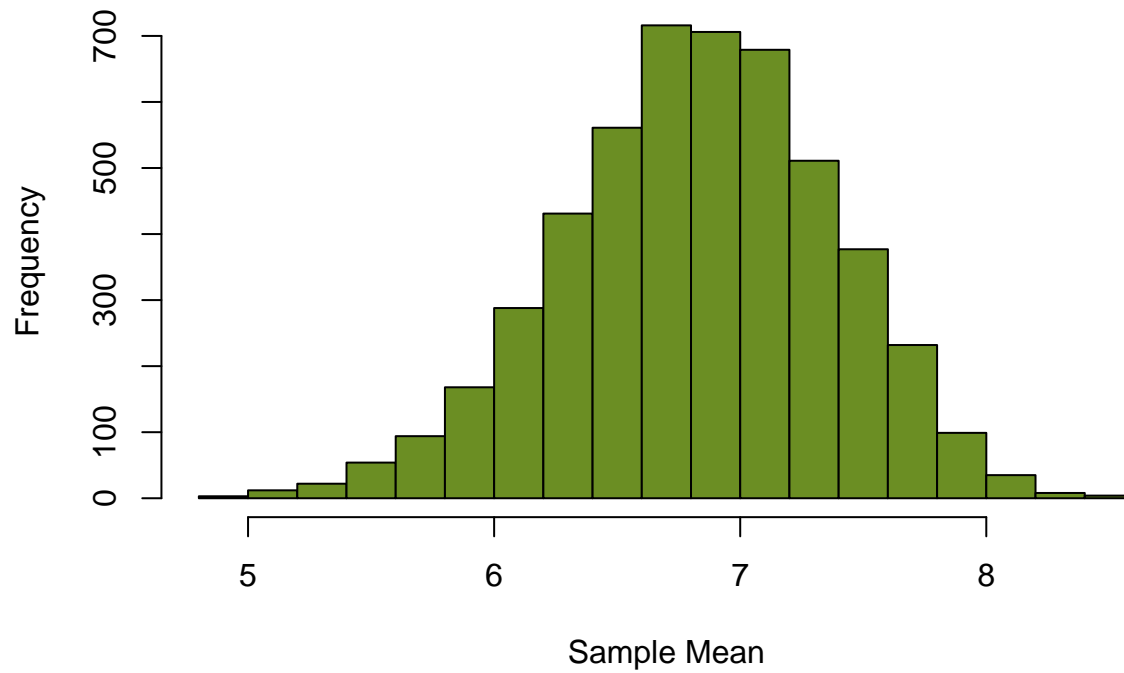
```
##      2.5%    97.5%
## 5.733333 7.866667
```

```
quantile(s_storage, c(0.025, 0.975))
```

```
##      2.5%    97.5%
## 1.125463 2.825142
```

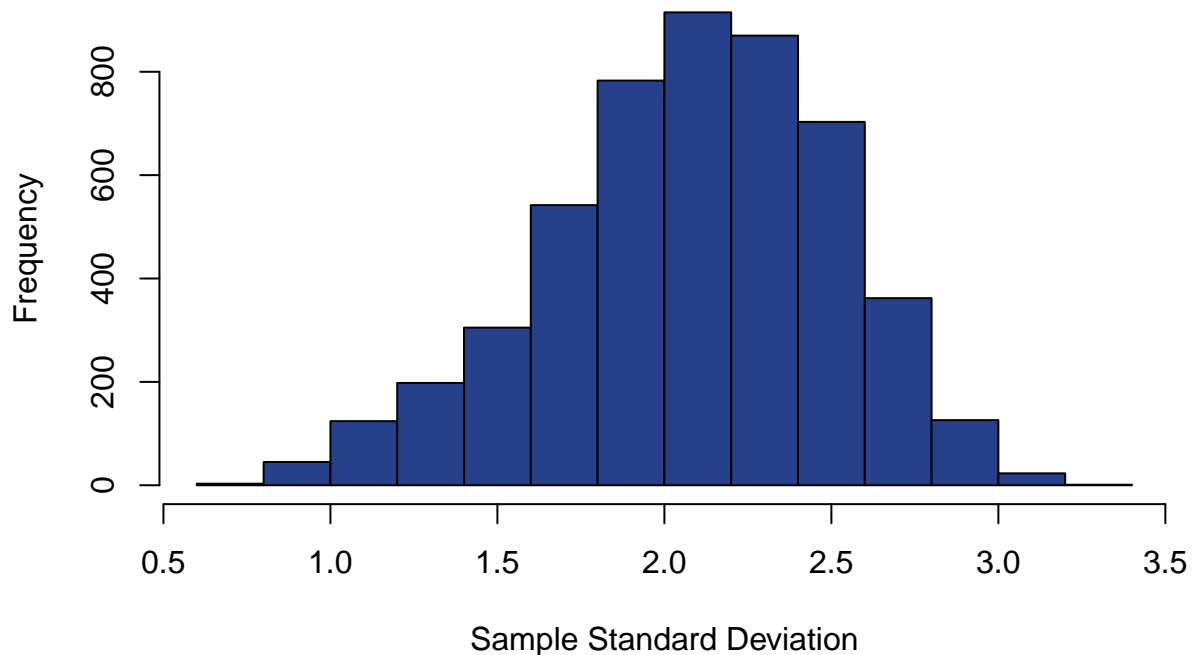
```
hist(xbar_storage, main='Sampling Distribution of Mean Neonatal APGAR Scores',
     xlab='Sample Mean', col='olivedrab')
```

Sampling Distribution of Mean Neonatal APGAR Scores



```
hist(s_storage, main='Sampling Distribution of Standard Deviation of Neonatal APGAR Scores',  
     xlab='Sample Standard Deviation', col='royalblue4')
```

Sampling Distribution of Standard Deviation of Neonatal APGAR Score



Part b)

Create a 95% bootstrap confidence interval for U , the Mann-Whitney statistic.

```
Ustorage <- rep(NA, B)
n1 <- length(apgar1)
n2 <- length(apgar2)

for (i in 1:B) {
  scores1<- sample(apgar1, size=n1, replace=TRUE)
  scores2 <- sample(apgar2, size=n2, replace=TRUE)
  scores <- data.frame("Method"=c(rep("1", n1), rep("2", n2)), "Score"=c(scores1,scores2))
  ranks <- rank(scores$Score)
  scores$Ranks <- ranks
  R1 <- sum(scores$Ranks[scores$Method == "1"])
  R2 <- sum(scores$Ranks[scores$Method == "2"])
  U1 <- n1*n2 + n1*(n1+1)/2 - R1
  U2 <- n1*n2 + n2*(n2+1)/2 - R2
  U <- min(U1, U2)
  Ustorage[i] <- U
}

quantile(Ustorage, c(0.025, 0.975))
```

```
## 2.5% 97.5%
```

```
## 1.0 22.5
```

Consistency

Show that \bar{x} and s^2 are both consistent for λ in a Poisson distribution.

```
NMC <- 5000
x <- rep(NA, NMC)
lambda <- 6.2
x[1] <- rpois(1, lambda)

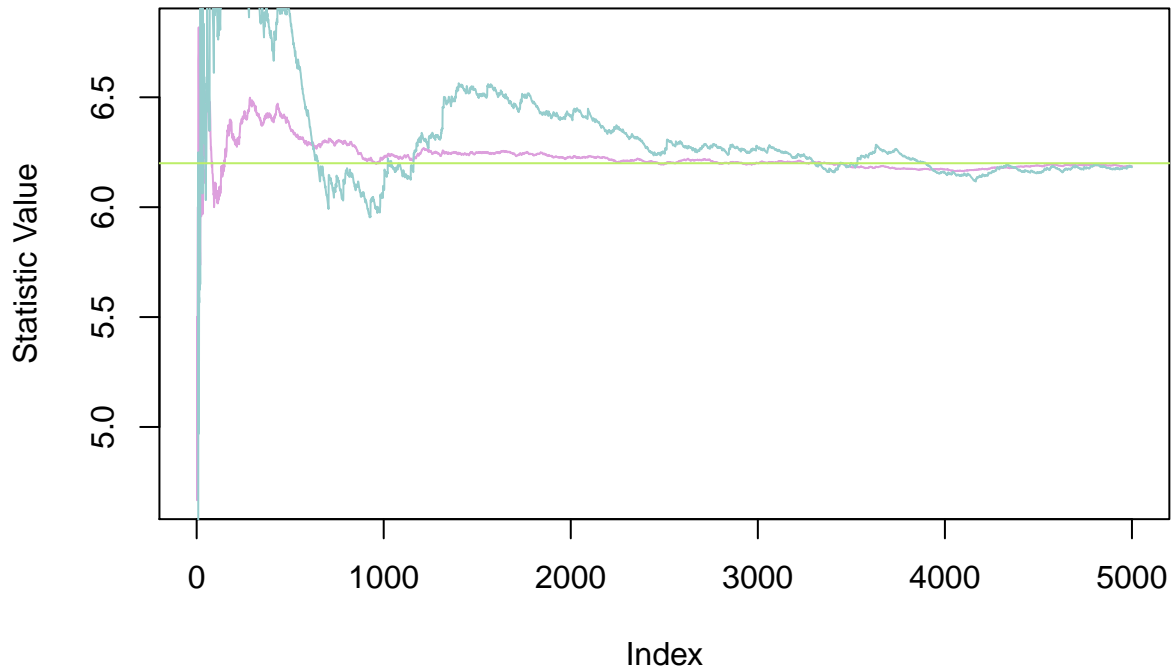
xbars <- rep(NA, NMC)
s2s <- rep(NA, NMC)

xbars[1] <- mean(x)
s2s <- 0

for (i in 2:NMC) {
  x[i] <- rpois(1, lambda)
  xbars[i] <- mean(x[1:i])
  s2s[i] <- var(x[1:i])
}

plot(xbars, type='l', main='Consistency of Sample Mean and Sample Variance', col='plum',
      ylab='Statistic Value')
lines(s2s, type='l', col='paleturquoise3')
abline(h=lambda, col="darkolivegreen2")
```

Consistency of Sample Mean and Sample Variance



Power and Type II Error

Consider two samples A and B from Bernoulli populations with parameters p_A and p_B and sample sizes n_A and n_B . With MC simulation and by trial and error, propose ps , ns , and $alpha$ such that you can have a power of about 80% of correctly rejecting $H_0 : p_A = p_B$ in favor of $H_A : p_A > p_B$.

```
pA <- 0.68
pB <- 0.41
nA <- 40
nB <- 37
alpha <- 0.05
pvalue_storage <- rep(NA, NMC)

for (i in 1:NMC) {
  sampleA <- rbinom(1, nA, pA)
  sampleB <- rbinom(1, nB, pB)
  phatA <- sampleA/nA
  phatB <- sampleB/nB
  zobs <- (phatA-phatB) / sqrt(phatA*(1-phatA)/nA + phatB*(1-phatB)/nB)
  pvalue_storage[i] <- pnorm(zobs, 0, 1, lower.tail=FALSE)
}

mean(pvalue_storage<alpha)
```

```
## [1] 0.8048
```