# Problem Set 3 – MATH392

*Theodore Dounias*

*2/11/2018*

### 4.8

```r
z <- (4.6 - 6)/(sqrt(.5))

p_hat <- pnorm(z, 0, 1)

p_hat
```

```
## [1] 0.02385744
```

### 4.9

I will first calculate mean and variance from the pdf. Then I will use the CLT approximation like above. For the mean:

$$E(X) = \int_2^6 xf(x)dx = \frac{3}{16}\int_2^6 (x^3 - 8x^2 + 16x)dx = \frac{3}{16}(\frac{1}{4}x^4 - \frac{8}{3}x^3 + 8x^2\Big|_2^6) \approx 4$$

To calculate the variance, we will first calculate E(X^2):

$$E(X^2) = \int_2^6 x^2 f(x)dx = \frac{3}{16}\int_2^6 (x^4 - 8x^3 + 16x^2)dx = \frac{3}{16}(\frac{1}{5}x^5 - 2x^4 + \frac{16}{3}x^3\Big|_2^6) \approx 18.4$$

We then use CLT approximation:

```r
z <- (4.2 - 4)/(sqrt((18.4 - 16)/244))

p_hat <- 1 - pnorm(z, 0, 1)

p_hat
```

```
## [1] 0.02186875
```

### 4.12

a. The expected value of the sample mean is equal to the population mean. In this case, therefore, it should be 10.

b.

```r
it <- 1000
n <- 30
means <- rep(0, 1000)

for(i in 1:it){
  means[i] <- mean(rexp(30, 0.1))
}

#This asks for a "proportion" and not a p-value, so I will use the following formula:
```

```r
prop_b <- sum(means >= 12)/it
prop_b
```

```
## [1] 0.153
```

c. "Unusual" is a hard word to wrap your head around. It certainly seems like this is a value that is not extreme, and so I would say that it is not unusual.

**4.13**

a. From a widely accepted result that the sum of normal distributions is normal, we have:

$$\bar{X} \sim N(20, (\frac{8}{\sqrt{10}})), \bar{Y} \sim N(16, (\frac{7}{\sqrt{15}}))$$

And so:

$$W \sim N(20 + 16, \sqrt{(\frac{7}{\sqrt{15}})^2 + (\frac{8}{\sqrt{10}})^2}) = N(36, 3.109^2)$$

b-c. I will use the code included in the book, I guess?

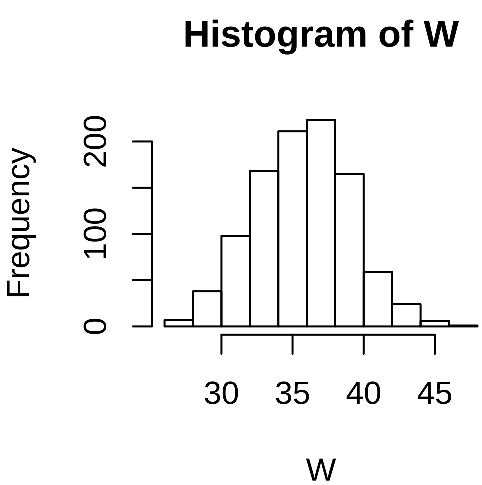
```r
W <- numeric(1000)
for(i in 1:1000){
  x <- rnorm(10, 20, 8)
  y <- rnorm(15, 16, 7)
  W[i] <- mean(x) + mean(y)
}

mean(W)
```

```
## [1] 35.72691
```

```r
sd(W)
```

```
## [1] 3.316211
```

```r
hist(W)
```



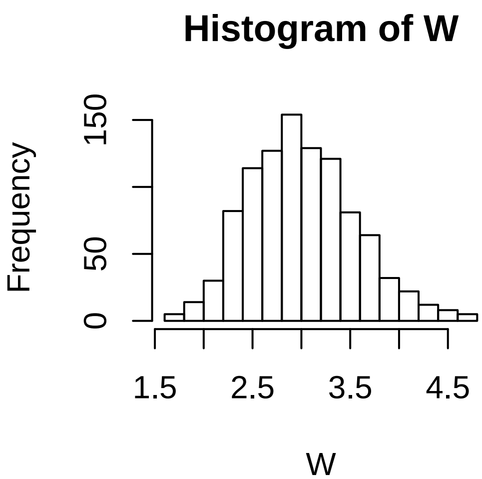

```r
mean(W < 40)
```

```
## [1] 0.91
```

**4.18**

a. For this I can just copy the revious code:

```
W <- numeric(1000)
for(i in 1:1000){
  x <- rexp(30, 1/3)
  W[i] <- mean(x)
}
hist(W)
```




b. Again, it is a common result that the sum of n iid exponential rv's follows a Gamma(L, n). Therefore we have:

```
k <- 30
lambda <- 1/3
#Theoretical mean:
1/lambda
```

```
## [1] 3
```

```
#Theoretical se
se <- 1/(lambda*sqrt(k))
se
```

```
## [1] 0.5477226
```

```
mean(W)
```

```
## [1] 3.002984
```

```
sd(W)
```

```
## [1] 0.5458038
```

c.

```
prob <- sum(W <= 3.5)/1000
prob
```

```
## [1] 0.817
```

d.

```
z <- (3.5 - 1/lambda)/se
```

```r
p_hat <- pnorm(z)

p_hat
```

```
## [1] 0.8193448
```

### 4.20

I will start from the cdf, and find the pdf by derivation to prove both of these expressions:

$$F_{min}(x) = 1 - P(min[X_1, ..., X_n] \geq x) =_{iid} 1 - P(X_1 \geq x)P(X_2 \geq x)...P(X_n \geq x) = 1 - (1 - F(x))^n$$

We can make this last step since $P(X_i \geq x) = 1 - P(X_i \leq x) = 1 - F_{X_i}(x)$, and the Xi's are iid.

And so we get:

$$f_{min}(x) = -n(1 - F(x))^{n-1}(-f(x)) = n(1 - F(x))^{n-1}f(x)$$

The second proof is exactly the same process, with the initial conversion being:

$$F_{max}(x) = P(max[X_1, ..., X_n] \leq x) =_{iid} P(X_1 \leq x)P(X_2 \leq x)...P(X_n \leq x) = F^n(x)$$

And so if we differentiate we get:

$$f_{max} = (n - 1)f(x)F^{n-1}(x)$$

### 4.21

We will first find the cdf of F–which is a weird way to name a distribution–, and then apply the formula proven above.

$$F(x) = \int_1^x 2/t^2 dt = -\frac{2}{t}\Big|_1^x = 2 - \frac{2}{x}$$

Applying the formula from 4.20 we have:

$$f_{max} = 2F(x)f(x) = 2(2 - \frac{2}{x})\frac{2}{x^2} = \frac{8}{x^2} - \frac{8}{x^3}$$

And so, to find the expected value we have:

$$E(X) = \int_1^2 xf(x)dx = \int_1^2 (\frac{8}{x} - \frac{8}{x^2})dx = (8ln(x) + \frac{8}{x})\Big|_1^2 = 1.55$$

### 5.2

A.

```r
#Can only happen in one permutation, so:
1/(4^4)
```

```
## [1] 0.00390625
```

B.

```r
1 - (3/4)^4
```

```
## [1] 0.6835938
```

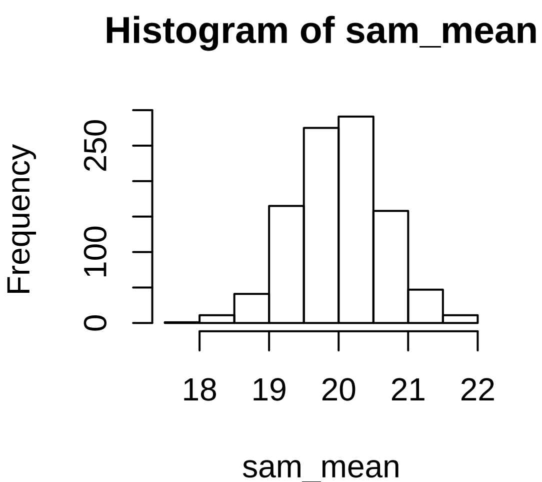
C.

```r
#This can be calculated with the binomial!
#Since order matters, we are looking for Binom(4, .25), with X = 2.
dbinom(2, 4, .25)
```

```
## [1] 0.2109375
```

**5.8**

```r
it <- 1000
sam_mean <- rep(0, 1000)
for(i in 1:1000){
  sam <- rgamma(200, 5, rate = 1/4)
  sam_mean[i] <- mean(sam)
}

hist(sam_mean)
```




This is approximatelly a normal with:

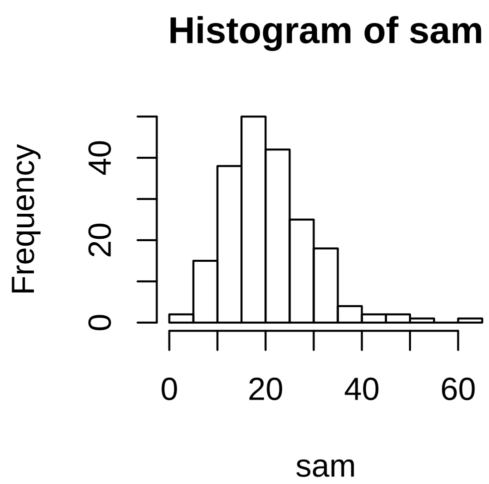
```r
mean(sam_mean)
```

```
## [1] 20.00855
```

```r
sd(sam_mean)
```

```
## [1] 0.6353073
```

```r
hist(sam)
```



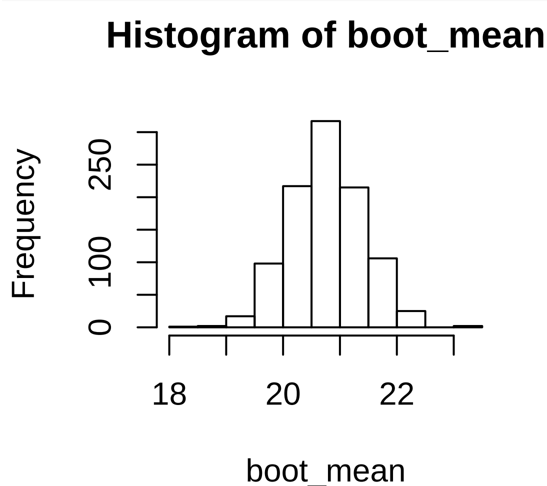

```
mean(sam)
```

```
## [1] 20.76638
```

```
sd(sam)
```

```
## [1] 9.01978
```

```
boot_mean <- rep(0, it)
for(i in 1:it){
  boot <- sample(sam, 200, replace = TRUE)
  boot_mean[i] <- mean(boot)
}

hist(boot_mean)
```




```
mean(boot_mean)
```

```
## [1] 20.76259
```

```
sd(boot_mean)
```

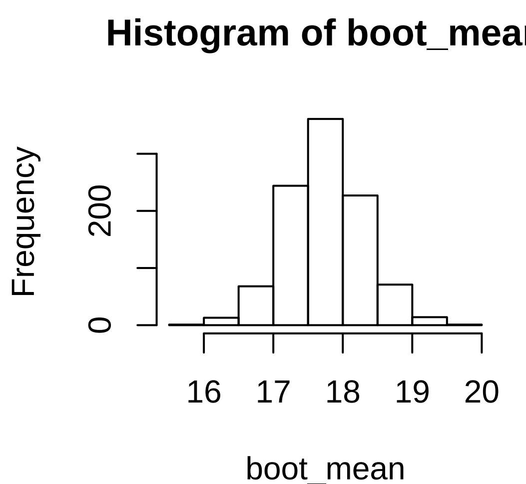
```
## [1] 0.636739
```

```
df <- data.frame(Mean <- c(20, mean(boot_mean)), SD <- c((4*sqrt(5))/sqrt(200), sd(boot_mean)))
names(df)[1]<- "Mean"
names(df)[2]<- "SD"
```

```r
print(df)
```

```
##       Mean        SD
## 1 20.00000 0.6324555
## 2 20.76259 0.6367390
```

```r
#n = 50
sam_mean <- rep(0, 1000)
for(i in 1:1000){
  sam <- rgamma(50, 5, rate = 1/4)
  sam_mean[i] <- mean(sam)
}

for(i in 1:it){
  boot <- sample(sam, 200, replace = TRUE)
  boot_mean[i] <- mean(boot)
}

hist(boot_mean)
```

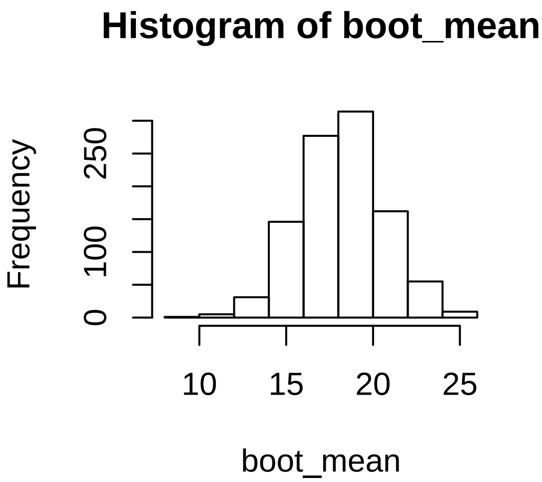



```r
df <- data.frame(Mean <- c(20, mean(boot_mean)), SD <- c((4*sqrt(5))/sqrt(50), sd(boot_mean)))
names(df)[1]<- "Mean"
names(df)[2]<- "SD"
print(df)
```

```
##       Mean        SD
## 1 20.00000 1.2649111
## 2 17.74205 0.5478785
```

```r
#n = 10
sam_mean <- rep(0, 1000)
for(i in 1:1000){
  sam <- rgamma(10, 5, rate = 1/4)
  sam_mean[i] <- mean(sam)
}

for(i in 1:it){
  boot <- sample(sam, 10, replace = TRUE)
  boot_mean[i] <- mean(boot)
}
```

```r
hist(boot_mean)
```



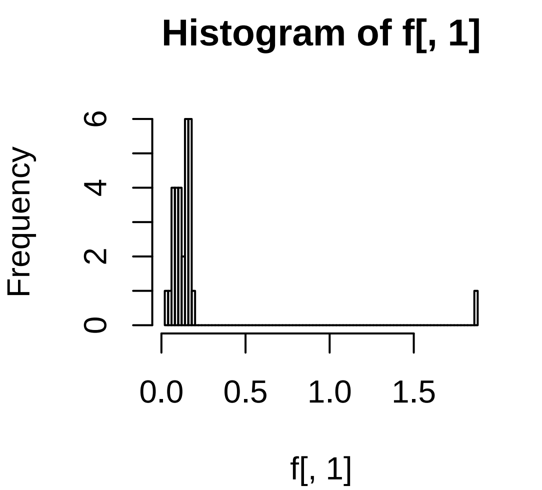

```r
df <- data.frame(Mean <- c(20, mean(boot_mean)), SD <- c((4*sqrt(5))/sqrt(10), sd(boot_mean)))
names(df)[1]<- "Mean"
names(df)[2]<- "SD"
print(df)
```

```
##       Mean       SD
## 1 20.00000 2.828427
## 2 18.22054 2.437896
```

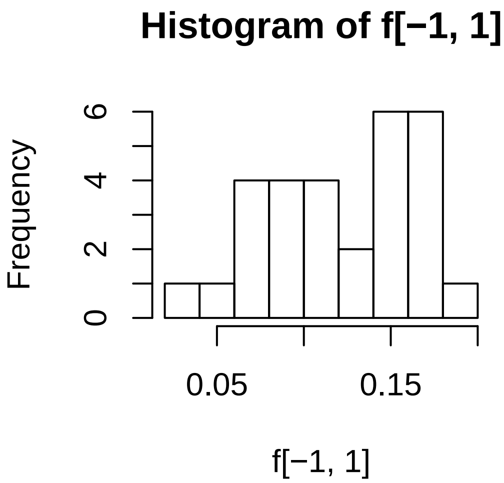
The bootstrap becomes less accurate at predicting the mean as the sample size becomes smaller. It still predicts the standard deviation fairly accuratelly, with some fluctuation.

**5.12**

```r
#The first histogram reveals that there is one outlier, which is removed for the second histogram. Afte
hist(f[,1], breaks=100)
```




```r
hist(f[-1,1])
```




```r
#With outlier
it <- 10000
boot_mean <- rep(0, it)
for(i in 1:it){
  boot <- sample(f[,1], replace = TRUE)
  boot_mean[i] <- mean(boot)
}

sd(boot_mean)
```

```
## [1] 0.05908881
```

```r
quantile(boot_mean, c(.025, .975))
```

```
##      2.5%     97.5%
## 0.1126333 0.3073667
```

```r
#Without outlier
boot_mean_1 <- rep(0, it)
for(i in 1:it){
  boot <- sample(f[-1,1], replace = TRUE)
  boot_mean_1[i] <- mean(boot)
}

sd(boot_mean_1)
```

```
## [1] 0.007858742
```

```r
quantile(boot_mean_1, c(.025, .975))
```

```
##      2.5%     97.5%
## 0.1079655 0.1386897
```

Removing the outlier significantly reduced the bootstraped mean's standard error, as well as the upper bound (and the lower bound, but less) of the confidence interval.