Qualifying Exam 2019

Exam #7

6/1/2019

Question 1

a) Number of meals involving fish as a positive test

```
# epiR package for calculating sensitivity and specificity
sensspec0 <- epi.tests(ctable0)
sensspec1 <- epi.tests(ctable1)
sensspec2 <- epi.tests(ctable2)
sensspec3 <- epi.tests(ctable3)
sensspec4 <- epi.tests(ctable4)
sensspec7 <- epi.tests(ctable7)
sensspec14 <- epi.tests(ctable14)
sensspec21 <- epi.tests(ctable21)
```

	Sensitivity	Specificity
>=0	100	0.0
>=1	100	8.0
>=2	100	19.2
>=3	100	28.0
>=4	90	28.8
>=7	70	36.8
>=14	30	89.6
>=21	30	93.6

b) Appropriate thresholds

Sensitivity refers to the true positive rate, or the probability that a test will rule in disease correctly. Specificity indicates the true negative rate, or the probability that a test will correctly rule out disease. Therefore, the probability of a false negative is 100 - sensitivity and the the false positive rate is 100 - specificity.

i. True positives

If we want to maximize true positives while minimizing false positives, the optimal threshold is the one with the highest sensitivity and lowest 100 - specificity. A threshold of >= 3 meals per week including fish would provide a 100% true positive rate and a 72% false negative rate.

ii. True negatives

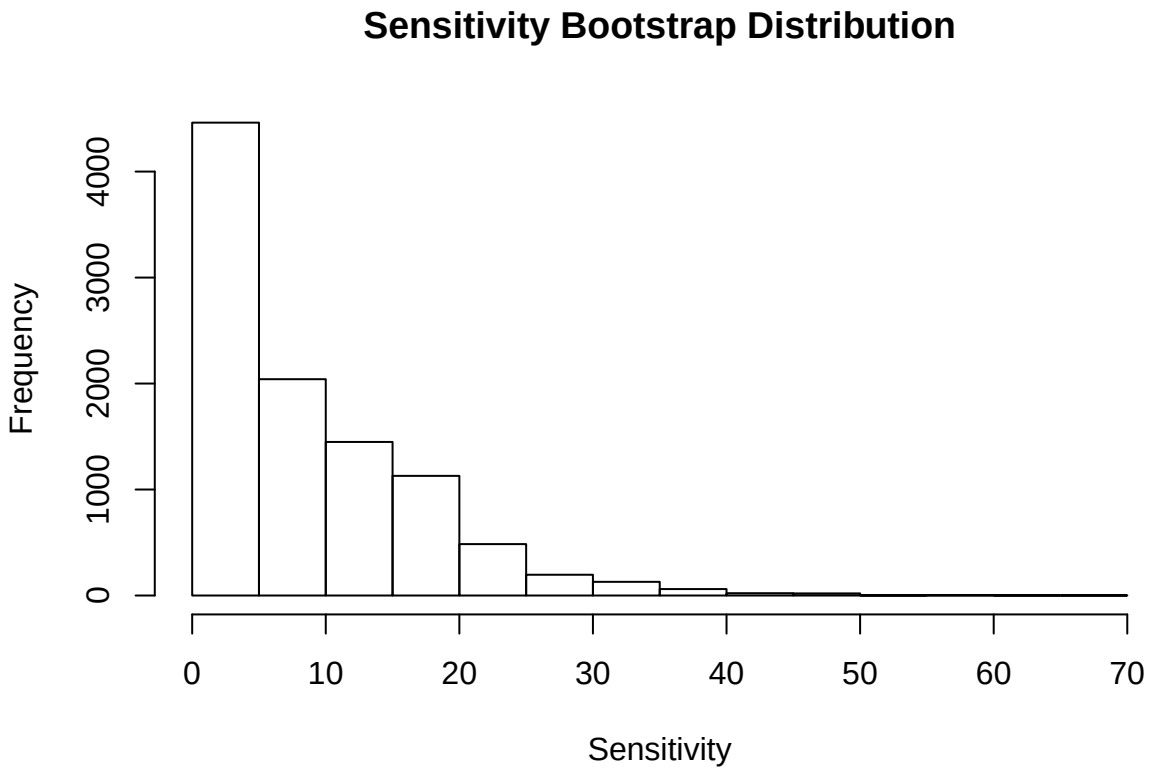
Maximizing true negatives first and then true positives requires choosing the test with highest specificity and highest sensitivity. In this case a threshold of >= 21 meals including fish per week would provide a true negative detection rate of 93.6% and a true positive rate of 30%.

c) Bootstrap sampling for >= 21 meals threshold

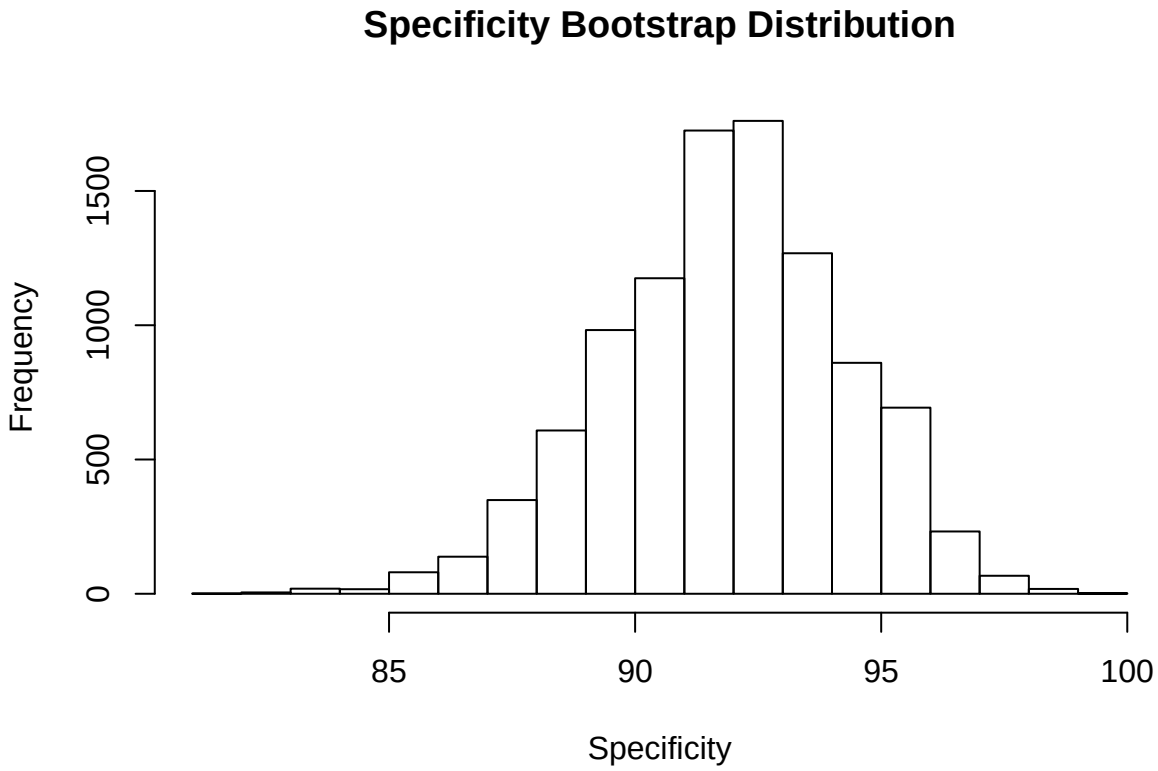
```
# Vector for storing results
set.seed(1234)
B <- 10000
sens_results <- numeric(B)
spec_results <- numeric(B)
# Loop
for (i in 1:B) {
  meals <- sample(fish$fishmlwk,replace = T)
  meals <- ifelse(meals >= 21,1,0)
  response <- sample(fish$MeHg,replace = T)
  response <- ifelse(response >= 8,1,0)
  table <- table(factor(meals,levels=1:0),factor(response,levels=1:0))
  sens_results[i] <- (table[1,1]/sum(table[,1])) * 100
  spec_results[i] <- (table[2,2]/sum(table[,2])) * 100
}
```

i. Plots

```
# Plots
hist(sens_results,main = "Sensitivity Bootstrap Distribution",xlab = "Sensitivity")
```



```r
hist(spec_results,main = "Specificity Bootstrap Distribution",xlab = "Specificity")
```



ii. Mean, SE, and Bias From Bootstrap Distributions

	Mean	Standard Error	Bias
Sensitivity	8.16521	0.0922186	-21.83479
Specificity	91.87575	0.0240551	-1.72425

iii. 90% Bootstrap and Normal Percentile Confidence Intervals

```r
# Sensitivity
# Normal percentiles
L <- mean(sens_results) - (1.645 * sd(sens_results))
L
```

```
## [1] -7.004749
```

```r
U <- mean(sens_results) + (1.645 * sd(sens_results))
U
```

```
## [1] 23.33517
```

```r
# Coverage
sum(sens_results < L)/B
```

```
## [1] 0
```

```r
sum(sens_results > U)/B
```

```
## [1] 0.0671
```

```r
# Bootstrap percentiles
quantile(sens_results,c(0.05,0.95))
```

```
##  5% 95%
##   0  25
```

The bootstrap distribution for sensitivity is not at all normal. The 90% confidence interval for this distribution using normal percentiles is (-7.00%,23.34%), which does not make sense as sensitivity cannot be negative. Also, none of the bootstrap values were below the lower limit (again, because this is impossible), when we'd expect that 5% would be for a normal distribution. So in this case it would probably be better to use the bootstrap confidence interval (0%,25%).

```r
# Specificity
# Normal percentiles
Lc <- mean(spec_results) - (1.645 * sd(spec_results))
Lc
```

```
## [1] 87.91868
```

```r
Uc <- mean(spec_results) + (1.645 * sd(spec_results))
Uc
```

```
## [1] 95.83282
```

```r
# Coverage
sum(spec_results < Lc)/B
```

```
## [1] 0.0558
```

```r
sum(spec_results > Uc)/B
```

```
## [1] 0.049
```

```r
# Bootstrap percentiles
quantile(spec_results,c(0.05,0.95))
```

```
##       5%      95%
## 87.80488 95.79832
```

The bootstrap distribution for specificity appears to be much closer to normal than sensitivity. The 90% normal percentile confidence interval is (87.92%,95.83%), which matches the bootstrap confidence interval very closely (87.80%,95.80%). Also, approximately 5% percent of the bootstrap values were below the lower limit and above the upper limit, which is what we would expect from a normal distribution.

d. 90% Confidence Intervals Using Exact and Asymptotic Methods

i. Sensitivity

Clopper-Pearson Method

$$\hat{p} = \frac{3}{10}$$

$$\text{CI} = \left(\frac{x}{x + (n - x + 1)F_{1-\frac{\alpha}{2};2(n-x+1),2x}}, \frac{(x+1)F_{1-\frac{\alpha}{2};2(x+1),2(n-x)}}{(n-x) + (x+1)F_{1-\frac{\alpha}{2};2(x+1),2(n-x)}}\right)$$

```
n <- sum(ctable21[,1])
x <- ctable21[1,1]
L <- x/(x+((n-x+1)*qf(0.95,(2*(n-x+1)),2*x))) * 100
L
```

```
## [1] 8.726443
```

```
U <- (x+1)*qf(0.95,(2*(x+1)),2*(n-x))/((n-x)+(x+1)*qf(0.95,(2*(x+1)),2*(n-x))) * 100
U
```

```
## [1] 60.66242
```

2. Simple Asymptotic (Normal Approximation to the Binomial Distribution)

$$\hat{p} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

```
n <- sum(ctable21[,1])
phat <- 0.3
L <- (phat - qnorm(0.95)*sqrt((phat*(1-phat))/n))*100
L
```

```
## [1] 6.163806
```

```
U <- (phat + qnorm(0.95)*sqrt((phat*(1-phat))/n))*100
U
```

```
## [1] 53.83619
```

ii. Specificity

1. Clopper-Pearson Method

$$\hat{p} = \frac{117}{125}$$

$$\text{CI} = (\frac{x}{x+(n-x+1)F_{1-\frac{\alpha}{2};2(n-x+1),2x}}, \frac{(x+1)F_{1-\frac{\alpha}{2};2(x+1),2(n-x)}}{(n-x)+(x+1)F_{1-\frac{\alpha}{2};2(x+1),2(n-x)}})$$

```
n <- sum(ctable21[,2])
x <- ctable21[2,2]
L <- x/(x+((n-x+1)*qf(0.95,(2*(n-x+1)),2*x))) * 100
L
```

```
## [1] 88.74873
```

```
U <- (x+1)*qf(0.95,(2*(x+1)),2*(n-x))/((n-x)+(x+1)*qf(0.95,(2*(x+1)),2*(n-x))) * 100
U
```

```
## [1] 96.77588
```

2. Simple Asymptotic (Normal Approximation to the Binomial Distribution)

$$\hat{p} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

```
n <- sum(ctable21[,2])
phat <- 0.936
L <- (phat - qnorm(0.95)*sqrt((phat*(1-phat))/n))*100
L
```

```
## [1] 89.99919
```

```
U <- (phat + qnorm(0.95)*sqrt((phat*(1-phat))/n))*100
U
```

```
## [1] 97.20081
```

The Clopper-Pearson CI for sensitivity is (8.73%,60.66%). The simple asymptotic CI for sensitivity is (6.16%,53.84%). The Clopper-Pearson CI for specificity is (88.75%,96.78%). The simple asymptotic CI for specificity is (90.00%,97.20%).

In general, the normal approximation works best for large sample sizes. Although as a general rule of thumb the Central Limit Theorem applies to sample sizes over 30, the boostrap distribution of sensitivity was not normally distributed so I would use the exact confidence interval in this case.

Confidence intervals are essentially the range for a parameter that is consistent with the data. So based on the exact confidence intervals, if we were to repeat this experiment many times, sensitivity for this test would be between 8.73% and 60.66% in 90% of those experiments.

e. Linear Regression

i. Model Equation

$$\hat{MeHg} = \hat{\beta}_0 + \hat{\beta}_1 X_{\text{fisherman}} + \hat{\beta}_2 X_{\text{fish meals per week}} + \hat{\beta}_3 X_{\text{fish parts=1}} + \hat{\beta}_4 X_{\text{fish parts=2}} + \hat{\beta}_5 X_{\text{fish parts=3}}$$

In the model above, $\hat{\beta}_1$ is the estimate for the effect of being a fisherman on mercury levels. $\hat{\beta}_2$ is the estimated effect of the number of fish meals per week on mercury levels. In this model we are treating the number of fish meals per week as continuous. $\hat{\beta}_3$,$\hat{\beta}_4$, and $\hat{\beta}_5$ are the estimated effect of eating muscle tissue only, muscle tissue and sometimes the whole fish, or the whole fish (respectively). $\hat{\beta}_0$, the intercept, is the average mercury level for someone who is not a fisherman, eats 0 fish meals per week, and do not consume any fish parts.

ii. Results

```
lin_mod <- lm(MeHg ~ factor(fisherman)+fishmlwk+factor(fishpart),data = fish)
results <- as.data.frame(summary(lin_mod)$coefficients)
rownames(results) <- c("Intercept","Fisherman = Yes",
                       "Fish Meals per Week","Fish Part = Muscle",
                       "Fish Part = Muscle and Whole",
                       "Fish Part = Whole")
kable(results)
```

	Estimate	Std. Error	t value	Pr(>|t|)
Intercept	0.9040000	0.8420135	1.0736170	0.2849986
Fisherman = Yes	0.2464962	0.7417227	0.3323293	0.7401801
Fish Meals per Week	0.0964710	0.0568348	1.6973942	0.0920335
Fish Part = Muscle	3.0608992	1.0782386	2.8387957	0.0052624
Fish Part = Muscle and Whole	1.6757469	1.0186581	1.6450533	0.1023934
Fish Part = Whole	3.0091672	1.3660412	2.2028378	0.0293821

iii. Summary