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

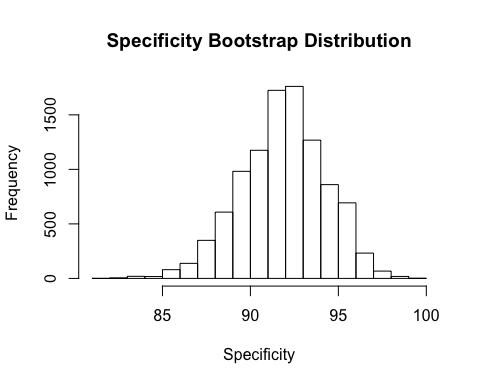
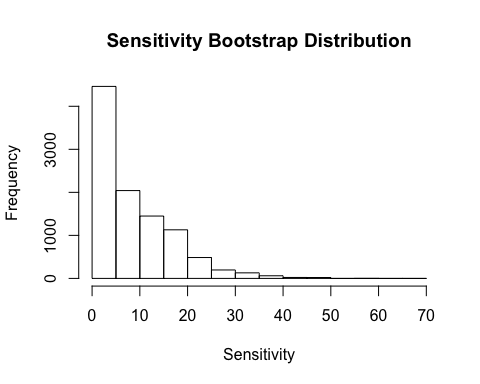
If we want to maximize true positives while minimizing false positives, the optimal threshold is the one with the highest sensitivity and lowest 100 - specificity. A threshold of >= 3 meals per week including fish would provide a 100% true positive rate and a 72% false negative rate.

### 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

### i. Plots



### 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

|  |  |  |  |
| --- | --- | --- | --- |
|  | Normal Percentile | Coverage | Bootstrap CI |
| Sensitivity | -7%, 23.34% | 0%, 6.71% | 0%, 25% |
| Specificity | 87.92%, 95.83% | 5.58%, 4.9% | 87.8%, 95.8% |

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%).

The bootstrap distribution for specificity appears to be much closer to normal than for 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 in each tail, which is what we would expect from a normal distribution.

## d. 90% Confidence Intervals Using Exact and Asymptotic Methods

|  |  |  |
| --- | --- | --- |
|  | Clopper-Pearson | Simple Asymptotic |
| Sensitivity | 8.73%, 60.66% | 6.16%, 53.84% |
| Specificity | 88.75%, 96.78% | 90%, 97.2% |

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

In the model above, is the estimate for the effect of being a fisherman on mercury levels. 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. ,, and are the estimated effect of eating muscle tissue only, muscle tissue and sometimes the whole fish, or the whole fish (respectively). , 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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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

On average, being a fisherman increases mercury levels by 0.246 (95% CI: -1.221,1.714), but this relationship is not statiostically significant (p = 0.740).

## f. Fishermen Who Eat 4 Meals of Whole Fish Each Week

### i. Average

On average, fishermen who eat 4 meals of whole fish each week will have a mercury level of 4.546 (95% CI: 3.071,6.019).

### ii. Individual

An individual fisherman who eats 4 meals of whole fish each week will have a mercury level of 4.546 (95% CI: -0.011,9.102).

### iii. Prediction Interval vs. Confidence interval

The confidence interval above gives us information about the average mercury level for fishermen who eat 4 meals of whole fish each week in the current sample. However, the prediction interval refers to the mercury level for a theoretical new study participant. So because we are trying to make inference about a broader population, we need to account for some uncertainty in our estimators, which results in a wider interval.

# Question 2

## a.

What restriction allows it to be identified? Betas are the restricted parameters.

Calculate the log likelihood of Yi:

This is a probit regression model, so we can use glm() to calculate the coefficient estimates and their standard error:

# Contamination as binary variable  
pcbs$Yi <- ifelse(pcbs$contam.lev >= 0.001,1,0)  
# Probit model  
probit\_mod <-   
 glm(Yi ~ factor(location), family = binomial(link = "probit"), data = pcbs)  
kable(summary(probit\_mod)$coefficients)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Estimate | Std. Error | z value | Pr(>|z|) |
| (Intercept) | 0.7215223 | 0.2365641 | 3.050007 | 0.0022884 |
| factor(location)Niagara | 1.1679877 | 0.4933294 | 2.367561 | 0.0179058 |

## b.

# Question 3

cat\_mod <- lmer(logvl~trt\*week+(1|pid),data=hiv)  
cat\_mod <- lme(logvl~trt\*week,random = ~1|pid,data=hiv)  
summary(cat\_mod)$tTable

## Value Std.Error DF t-value p-value  
## (Intercept) 4.013867591 0.0772666570 2910 51.948250 0.000000e+00  
## trt -0.837790042 0.1087869774 222 -7.701198 4.417947e-13  
## week 0.005365891 0.0008358772 2910 6.419473 1.591148e-10  
## trt:week 0.001361022 0.0011768667 2910 1.156479 2.475802e-01

# Appendix

## Question 1

### Import and format data

# Import datasets  
fish <- read.csv("/Users/timvigers/Documents/School/Qualifying Exams/2019 MS/fishermen\_mercury.csv")  
pcbs <- read.csv("/Users/timvigers/Documents/School/Qualifying Exams/2019 MS/pcb\_conc.csv")  
hiv <- read.csv("/Users/timvigers/Documents/School/Qualifying Exams/2019 MS/hiv\_comp\_tall.csv",header = F)  
colnames(hiv) <- c("pid","trt","week","logvl")

### Create dummy variables for fish data

# Create indicator variables and response  
fish$meal0 <- ifelse(fish$fishmlwk >= 0,1,0)  
fish$meal1 <- ifelse(fish$fishmlwk >= 1,1,0)  
fish$meal2 <- ifelse(fish$fishmlwk >= 2,1,0)  
fish$meal3 <- ifelse(fish$fishmlwk >= 3,1,0)  
fish$meal4 <- ifelse(fish$fishmlwk >= 4,1,0)  
fish$meal7 <- ifelse(fish$fishmlwk >= 7,1,0)  
fish$meal14 <- ifelse(fish$fishmlwk >= 14,1,0)  
fish$meal21 <- ifelse(fish$fishmlwk >= 21,1,0)  
fish$response <- ifelse(fish$MeHg >= 8,1,0)  
# Create contingency tables  
ctable0 <- table(factor(fish$meal0,levels=1:0),factor(fish$response,levels=1:0))  
ctable1 <- table(factor(fish$meal1,levels=1:0),factor(fish$response,levels=1:0))  
ctable2 <- table(factor(fish$meal2,levels=1:0),factor(fish$response,levels=1:0))  
ctable3 <- table(factor(fish$meal3,levels=1:0),factor(fish$response,levels=1:0))  
ctable4 <- table(factor(fish$meal4,levels=1:0),factor(fish$response,levels=1:0))  
ctable7 <- table(factor(fish$meal7,levels=1:0),factor(fish$response,levels=1:0))  
ctable14 <- table(factor(fish$meal14,levels=1:0),factor(fish$response,levels=1:0))  
ctable21 <- table(factor(fish$meal21,levels=1:0),factor(fish$response,levels=1:0))  
# Make results table  
sens\_spec\_results <- as.data.frame(matrix(ncol = 2,nrow = 8))  
colnames(sens\_spec\_results) <- c("Sensitivity","Specificity")  
rownames(sens\_spec\_results) <- c(">=0",">=1",">=2",">=3",">=4",">=7",">=14",">=21")

### Format sensitivity and specificity results

# Format results  
sens\_spec\_results[">=0","Specificity"] <-   
 round(sensspec0$elements$specificity$est\*100,1)  
sens\_spec\_results[">=0","Sensitivity"] <-   
 round(sensspec0$elements$sensitivity$est\*100,1)  
sens\_spec\_results[">=1","Specificity"] <-   
 round(sensspec1$elements$specificity$est\*100,1)  
sens\_spec\_results[">=1","Sensitivity"] <-   
 round(sensspec1$elements$sensitivity$est\*100,1)  
sens\_spec\_results[">=2","Specificity"] <-   
 round(sensspec2$elements$specificity$est\*100,1)  
sens\_spec\_results[">=2","Sensitivity"] <-   
 round(sensspec2$elements$sensitivity$est\*100,1)  
sens\_spec\_results[">=3","Specificity"] <-   
 round(sensspec3$elements$specificity$est\*100,1)  
sens\_spec\_results[">=3","Sensitivity"] <-   
 round(sensspec3$elements$sensitivity$est\*100,1)  
sens\_spec\_results[">=4","Specificity"] <-   
 round(sensspec4$elements$specificity$est\*100,1)  
sens\_spec\_results[">=4","Sensitivity"] <-   
 round(sensspec4$elements$sensitivity$est\*100,1)  
sens\_spec\_results[">=7","Specificity"] <-   
 round(sensspec7$elements$specificity$est\*100,1)  
sens\_spec\_results[">=7","Sensitivity"] <-   
 round(sensspec7$elements$sensitivity$est\*100,1)  
sens\_spec\_results[">=14","Specificity"] <-   
 round(sensspec14$elements$specificity$est\*100,1)  
sens\_spec\_results[">=14","Sensitivity"] <-   
 round(sensspec14$elements$sensitivity$est\*100,1)  
sens\_spec\_results[">=21","Specificity"] <-   
 round(sensspec21$elements$specificity$est\*100,1)  
sens\_spec\_results[">=21","Sensitivity"] <-   
 round(sensspec21$elements$sensitivity$est\*100,1)

### Bootstrap

# 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  
}

### Mean, SE, and Bias for Bootstrap Distributions

boot\_results <- as.data.frame(matrix(ncol = 3,nrow = 2))  
colnames(boot\_results) <- c("Mean","Standard Error","Bias")  
rownames(boot\_results) <- c("Sensitivity","Specificity")  
# Sensitivity  
boot\_results["Sensitivity","Mean"] <- mean(sens\_results)  
boot\_results["Sensitivity","Standard Error"] <-   
 sd(sens\_results)/sqrt(length(sens\_results))  
boot\_results["Sensitivity","Bias"] <-   
 mean(sens\_results) - sensspec21$elements$sensitivity$est\*100  
# Specificity  
boot\_results["Specificity","Mean"] <- mean(spec\_results)  
boot\_results["Specificity","Standard Error"] <-   
 sd(spec\_results)/sqrt(length(spec\_results))  
boot\_results["Specificity","Bias"] <-   
 mean(spec\_results) - sensspec21$elements$specificity$est\*100

### Bootstrap and Normal Percentile Confidence Intervals

# Sensitivity  
# Normal percentiles  
L <- mean(sens\_results) - (1.645 \* sd(sens\_results))  
U <- mean(sens\_results) + (1.645 \* sd(sens\_results))  
# Specificity  
# Normal percentiles  
Lc <- mean(spec\_results) - (1.645 \* sd(spec\_results))  
Uc <- mean(spec\_results) + (1.645 \* sd(spec\_results))  
# Results table  
results <- as.data.frame(matrix(ncol = 3,nrow = 2))  
rownames(results) <- c("Sensitivity","Specificity")  
colnames(results) <- c("Normal Percentile","Coverage","Bootstrap CI")  
L <- round(L,2)  
U <- round(U,2)  
Lc <- round(Lc,2)  
Uc <- round(Uc,2)  
results["Sensitivity",] <-   
 c(paste0(L,"%, ",U,"%"),  
 paste0(round(sum(sens\_results < L)/B \* 100,2),"%, ",  
 round(sum(sens\_results > U)/B \* 100,2),"%"),  
 paste0(paste(round(quantile(sens\_results,c(0.05,0.95)),2),collapse = "%, "),"%"))  
results["Specificity",] <-   
 c(paste0(Lc,"%, ",Uc,"%"),  
 paste0(round(sum(spec\_results < Lc)/B \* 100,2),"%, ",  
 round(sum(spec\_results > Uc)/B \* 100,2),"%"),  
 paste0(paste(round(quantile(spec\_results,c(0.05,0.95)),2),collapse = "%, "),"%"))

### 90% Confidence Intervals Using Exact and Asymptotic Methods

# Sensitivity  
# Clopper-Pearson  
results <- as.data.frame(matrix(ncol = 2,nrow = 2))  
rownames(results) <- c("Sensitivity","Specificity")  
colnames(results) <- c("Clopper-Pearson","Simple Asymptotic")  
n <- sum(ctable21[,1])  
x <- ctable21[1,1]  
L <- x/(x+((n-x+1)\*qf(0.95,(2\*(n-x+1)),2\*x))) \* 100  
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  
results["Sensitivity",1] <- paste0(round(L,2),"%, ",round(U,2),"%")  
# Simple asymptotic  
n <- sum(ctable21[,1])  
phat <- 0.3  
L <- (phat - qnorm(0.95)\*sqrt((phat\*(1-phat))/n))\*100  
U <- (phat + qnorm(0.95)\*sqrt((phat\*(1-phat))/n))\*100  
results["Sensitivity",2] <- paste0(round(L,2),"%, ",round(U,2),"%")  
# Specificity  
# Clopper-Pearson  
n <- sum(ctable21[,2])  
x <- ctable21[2,2]  
L <- x/(x+((n-x+1)\*qf(0.95,(2\*(n-x+1)),2\*x))) \* 100  
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  
results["Specificity",1] <- paste0(round(L,2),"%, ",round(U,2),"%")  
# Simple asymptotic  
n <- sum(ctable21[,2])  
phat <- 0.936  
L <- (phat - qnorm(0.95)\*sqrt((phat\*(1-phat))/n))\*100  
U <- (phat + qnorm(0.95)\*sqrt((phat\*(1-phat))/n))\*100  
results["Specificity",2] <- paste0(round(L,2),"%, ",round(U,2),"%")

### Linear Model

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")

### Confidence and Prediction Intervals

Confidence

a <- matrix(c(1,1,4,0,0,1))  
b <- as.numeric(summary(lin\_mod)$coefficients[,1])  
yhat <- t(a)%\*%b  
mse <- mean(lin\_mod$residuals^2)  
t <- qt(0.1/2,133,lower.tail = F)  
x <- model.matrix(lin\_mod)  
L <- yhat - t\*sqrt(mse\*(t(a)%\*%(solve((t(x)%\*%x)))%\*%a))  
U <- yhat + t\*sqrt(mse\*(t(a)%\*%(solve((t(x)%\*%x)))%\*%a))

Prediction

a <- matrix(c(1,1,4,0,0,1))  
b <- as.numeric(summary(lin\_mod)$coefficients[,1])  
yhat <- t(a)%\*%b  
mse <- mean(lin\_mod$residuals^2)  
t <- qt(0.1/2,133,lower.tail = F)  
x <- model.matrix(lin\_mod)  
L <- yhat - t\*sqrt(mse\*(1+(t(a)%\*%(solve((t(x)%\*%x)))%\*%a)))  
U <- yhat + t\*sqrt(mse\*(1+(t(a)%\*%(solve((t(x)%\*%x)))%\*%a)))