Data-581 Lab 1

Dhun Sheth

2023-11-21

Question 1

Part A

```
SEcalculator <- function(x, FUN, N = 1000) {
  xResample <- numeric(N)
  for (i in 1:N) {
    xResample[i] <- FUN(sample(x, size = length(x), replace=TRUE))
    }
  sd(xResample)
}

changes <- diff(LakeHuron)
mean_standard_error_function <- SEcalculator(changes, mean)
print(mean_standard_error_function)
```

```
## [1] 0.0759293
```

```
mean_usual_standard_error <- sd(changes)/sqrt(length(changes))
print(mean_usual_standard_error)
```

```
## [1] 0.07605443
```

Part B

```
var_standard_error_function <- SEcalculator(changes, var)
print(var_standard_error_function)
```

```
## [1] 0.07842993
```

Question 2

```r
q90 <- function(x, N=1000){
  xResample <- numeric(N)
  for (i in 1:N) {
    xResample[i] <- quantile(sample(x, size = length(x), replace=TRUE), 0.9)
  }
  list(estimate = quantile(x, 0.9), SE = sd(xResample))
}

quant_90 <- q90(changes)
print(quant_90)
```

```
## $estimate
##   90%
## 0.962
##
## $SE
## [1] 0.1493935
```

Question 3

Part A

Because X1 and X2 are independent, their joint density is a product of their individual densities.

Likelihood function for p: $L(p) = \prod_{i=1}^{n}[p^(x1_i + x2_i) * (1-p)^(2 - x1_i - x2_i)]$

Part B

```r
p <- c(0.2, 0.7, 0.9)
x1 <- 1
x2 <- 0
pdf_vals <- ((p^x1)*((1-p)^(1-x1)))*((p^x2)*((1-p)^(1-x2)))
print(pdf_vals)
```

```
## [1] 0.16 0.21 0.09
```

Maximum likelihood estimate for p is 0.7, because that gives the largest likelihood function value.

Part C

For X1=1 and X2=0, the likelihood function simplifies to: $[p^1 * (1-p)^0] * [p^0 * (1-p)^1] = p^1 * (1-p)^1 = p - p^2$

then, $\frac{\partial}{\partial p} p - p^2 = -2p + 1$

then set this to 0 to find maximum: $-2p + 1 = 0 => p = 1/2$

Thus, maximum likelihood estimate of p, if p ranges from (0,1), would be p=0.5

Question 6

Part A

```
# help(p13.2)
```

Binary logistic regression makes sense because the response variable (y), home ownership is either a 0 or 1.

Part B, C

```
p13.glm <- glm(formula = y ~ x, family = binomial, data = p13.2)

print(p13.glm)
```

```
##
## Call:  glm(formula = y ~ x, family = binomial, data = p13.2)
##
## Coefficients:
## (Intercept)            x
##  -8.7395139    0.0002009
##
## Degrees of Freedom: 19 Total (i.e. Null);  18 Residual
## Null Deviance:       27.53
## Residual Deviance: 22.43     AIC: 26.43
```

```
print(is.list(p13.glm))
```

```
## [1] TRUE
```

Part D

logit(x) = -8.7395139 + 2.0090564×10^{-4} * x

Question 7

```
set.seed(123)
x <- c(12, 14, 14, 15, 18, 21, 25, 29, 32, 35)

#define function to return median
myMedian <- function(x,i){median(x[i])}

result <- boot(x, myMedian, 200)
print(result)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = x, statistic = myMedian, R = 200)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*     19.5  1.2375    4.555337
```

Boot SE: 4.56
SEcalculator SE: 4.34