MockLabTestSolution2024Script

WY

2024-09-30

Q1a

```
#If final answers wrong, award 2pt for correct function
ilogit <- function(z){

  return(1/(1+exp(-z)))

}

GenerateLogit <- function(n){

  SeqInit <- seq(1,n,1)
  probVec <- ilogit(-0.15 + 0.005*SeqInit)
  return(rbinom(n = n, size = 1, prob = probVec))

}

#Testing GenerateLogit function
set.seed(1)
vy <- GenerateLogit(20)
sum(vy)

## [1] 11

var(vy)

## [1] 0.2605263
```

Q1b

```
vx <- seq(1,20,1)
#If final answers wrong, award 1pt for glm command call
LogitMod <- glm(vy~vx, family=binomial)
summ <- summary(LogitMod)
print(summ)

##
## Call:
## glm(formula = vy ~ vx, family = binomial)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.08624    0.93220  -0.093    0.926
## vx           0.02744    0.07839   0.350    0.726
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27.526  on 19  degrees of freedom
## Residual deviance: 27.402  on 18  degrees of freedom
## AIC: 31.402
##
## Number of Fisher Scoring iterations: 4
```

```
#If final answers wrong, award 1pt if only one mistake in the below code.
c(coef(summ)[2,1] - qnorm(0.95)*sqrt(summ$cov.unscaled[2,2]), coef(summ)[2,1]
+ qnorm(0.95)*sqrt(summ$cov.unscaled[2,2]))
```

```
## [1] -0.1014906  0.1563778
```

Q1c

```
#If final answers wrong, award 2pt for correct function. Note that having
CIlowerVec and CIupperVec as function output is optional.
RepeatedCI <- function(n, M.reps){

  vx.perm <- seq(1,n,1)
  ContainTheta1 <- rep(0,M.reps)
  CIlowerVec <- rep(0,M.reps)
  CIupperVec <- rep(0,M.reps)
  for(i in 1:M.reps){

    set.seed(i)
    vy.temp <- GenerateLogit(n)
    LogitMod.temp <- glm(vy.temp~vx.perm, family=binomial)
    summ.temp <- summary(LogitMod.temp)
    CIlowerVec[i] <- coef(summ.temp)[2,1] -
qnorm(0.95)*sqrt(summ.temp$cov.unscaled[2,2])
    CIupperVec[i] <- coef(summ.temp)[2,1] +
qnorm(0.95)*sqrt(summ.temp$cov.unscaled[2,2])
    ContainTheta1[i] <-
as.numeric((CIlowerVec[i]<=0.005)*(CIupperVec[i]>=0.005))

  }

  return(list(ContainTheta1=ContainTheta1, CIlowerVec=CIlowerVec,
CIupperVec=CIupperVec))

}

CIresults.M5 <- RepeatedCI(n = 35, M.reps = 5)
mean(CIresults.M5$ContainTheta1)
```

```
## [1] 0.8
```

Q1d

```
#No partial credit for wrong command.
CIresults.M20 <- RepeatedCI(n = 35, M.reps = 20)
mean(CIresults.M20$ContainTheta1)

## [1] 0.85

CIresults.M50 <- RepeatedCI(n = 35, M.reps = 50)
mean(CIresults.M50$ContainTheta1)

## [1] 0.82

CIresults.M500 <- RepeatedCI(n = 35, M.reps = 500)
mean(CIresults.M500$ContainTheta1)

## [1] 0.89

CIresults.M5000 <- RepeatedCI(n = 35, M.reps = 5000)
mean(CIresults.M5000$ContainTheta1)

## [1] 0.9016
```

Q1e

```
#answer is just 0
```

Q2a

```
data(swiss)
#If final answer wrong, award 1pt for correct lm command.
mod1 <- lm(Fertility ~ . , data = swiss)
summary(mod1)

##
## Call:
## lm(formula = Fertility ~ ., data = swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2743  -5.2617   0.5032   4.1198  15.3213
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
## Agriculture      -0.17211    0.07030  -2.448  0.01873 *
## Examination      -0.25801    0.25388  -1.016  0.31546
## Education        -0.87094    0.18303  -4.758 2.43e-05 ***
## Catholic          0.10412    0.03526   2.953  0.00519 **
## Infant.Mortality  1.07705    0.38172   2.822  0.00734 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 7.165 on 41 degrees of freedom
## Multiple R-squared:  0.7067, Adjusted R-squared:  0.671
## F-statistic: 19.76 on 5 and 41 DF,  p-value: 5.594e-10

sum(swiss$Fertility^2) - sum(mod1$residuals^2)

## [1] 236311.9
```

Q2b

```
modFull <- lm(Fertility ~ . , data = swiss)
drop1(object = modFull, test = "F")

## Single term deletions
##
## Model:
## Fertility ~ Agriculture + Examination + Education + Catholic +
##     Infant.Mortality
##                  Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                        2105.0 190.69
## Agriculture       1    307.72 2412.8 195.10  5.9934  0.018727 *
## Examination       1     53.03 2158.1 189.86  1.0328  0.315462
## Education         1   1162.56 3267.6 209.36 22.6432 2.431e-05 ***
## Catholic          1    447.71 2552.8 197.75  8.7200  0.005190 **
## Infant.Mortality  1    408.75 2513.8 197.03  7.9612  0.007336 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

mod.no.Exam <- lm(Fertility ~ Agriculture + Education + Catholic +
Infant.Mortality , data = swiss)
drop1(object = mod.no.Exam, test = "F")

## Single term deletions
##
## Model:
## Fertility ~ Agriculture + Education + Catholic + Infant.Mortality
##                  Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                        2158.1 189.86
## Agriculture       1    264.18 2422.2 193.29  5.1413   0.02857 *
## Education         1   2249.97 4408.0 221.43 43.7886 5.140e-08 ***
## Catholic          1    956.57 3114.6 205.10 18.6165 9.503e-05 ***
## Infant.Mortality  1    409.81 2567.9 196.03  7.9757   0.00722 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Q2c

```
#If final answer is wrong, 2pt awarded for correct command.
anova(mod.no.Exam,modFull)

## Analysis of Variance Table
##
```

```
## Model 1: Fertility ~ Agriculture + Education + Catholic + Infant.Mortality
## Model 2: Fertility ~ Agriculture + Examination + Education + Catholic + 
##     Infant.Mortality
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     42 2158.1                           
## 2     41 2105.0  1    53.027 1.0328 0.3155
```