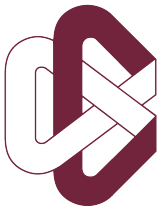

CIMAT

Centro de Investigación en Matemáticas
Unidad Monterrey

Cómputo Estadístico
Examen 1

Gustavo Hernández Angeles

10 de octubre de 2025

Índice

Ejercicio 1:

Se tiene la siguiente tabla donde se eligen varios niveles de ronquidos y se ponen en relación con una enfermedad cardíaca. Se toman como puntuaciones relativas de ronquidos los valores $\{0, 2, 4, 5\}$.

Ronquido	SI	NO	Proporción de SI
(0) Nunca	24	1355	0.017
(2) Ocasional	35	603	0.055
(4) Casi cada noche	21	192	0.099
(5) Cada noche	30	224	0.118

Ajuste un modelo logit y un modelo probit a estos datos e interprete los resultados. Compare los dos modelos y determine cuál es mejor.

Solución:

Primero construimos los datos en un dataframe de R.

```
ronq <- c(0, 2, 4, 5)
SI <- c(24, 35, 21, 30)
NO <- c(1355, 603, 192, 224)
df <- data.frame(ronq = ronq, SI = SI, NO = NO)
df$N <- df$SI + df$NO
df$prop <- df$SI / df$N
print(df)
```

```
##   ronq SI   NO    N       prop
## 1    0 24 1355 1379 0.01740392
## 2    2 35  603  638 0.05485893
## 3    4 21  192  213 0.09859155
## 4    5 30  224  254 0.11811024
```

Ajustamos ambos modelos (logit y probit) e imprimimos los resúmenes.

```
m_logit <- glm(cbind(SI, NO) ~ ronq, data = df, family = binomial(link = "logit"))
m_probit <- glm(cbind(SI, NO) ~ ronq, data = df, family = binomial(link = "probit"))
```

Veamos el resumen del modelo logit.

```
print(summary(m_logit))
```

```
##
## Call:
## glm(formula = cbind(SI, NO) ~ ronq, family = binomial(link = "logit"),
##     data = df)
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.86625    0.16621 -23.261  < 2e-16 ***
## ronq         0.39734    0.05001   7.945 1.94e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 65.9045  on 3  degrees of freedom
## Residual deviance:  2.8089  on 2  degrees of freedom
## AIC: 27.061
##
## Number of Fisher Scoring iterations: 4
```

Podemos observar que el coeficiente de la variable 'ronq' es positivo (0.3973) para el modelo logit, lo que indicaría que a medida que aumenta el nivel de ronquido, también aumenta la probabilidad de enfermedad cardíaca. Además, el intercepto es negativo (con magnitud 3.86), lo que sugiere que en ausencia de ronquidos (ronq = 0), la probabilidad de enfermedad cardíaca es baja. Ambas variables son estadísticamente significativas, contando con un p-valor muy cercano al 0.

El valor del intercepto en el modelo logit representa el logaritmo del odds (razón de probabilidades) de tener la enfermedad cardíaca cuando el nivel de ronquido es cero. En este caso, utilizando el valor del intercepto (-3.86), podemos calcular el odds como $e^{-3.86} \approx 0.021$. Esto indica que, cuando no hay ronquidos, la probabilidad de tener la enfermedad cardíaca es baja en comparación con no tenerla.

El valor del coeficiente de 'ronq' (0.3973) representa el cambio en el logaritmo del odds por cada unidad adicional en el nivel de ronquido. Para interpretar esto en términos de odds ratio, podemos calcular $e^{0.3973} \approx 1.487$. Esto significa que por cada aumento de una unidad en el nivel de ronquido, los odds de tener la enfermedad cardíaca aumentan en un factor de aproximadamente 1.487, lo que indica un aumento significativo en la probabilidad de enfermedad cardíaca con niveles más altos de ronquidos.

Ahora veamos el resumen del modelo probit.

```
print(summary(m_probit))
```

```
##
## Call:
## glm(formula = cbind(SI, NO) ~ ronq, family = binomial(link = "probit"),
##     data = df)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.06055    0.07017 -29.367  < 2e-16 ***
## ronq         0.18777    0.02348   7.997 1.28e-15 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 65.9045  on 3  degrees of freedom
## Residual deviance:  1.8716  on 2  degrees of freedom
## AIC: 26.124
##
## Number of Fisher Scoring iterations: 4
```

Nuevamente, observamos que el coeficiente de la variable 'ronq' es positivo (0.1877) para el modelo probit, lo que indica que a medida que aumenta el nivel de ronquido, también aumenta la probabilidad de enfermedad cardíaca. El intercepto es negativo (con magnitud 2.06), sugiriendo una baja probabilidad de enfermedad cardíaca cuando no hay ronquidos. Al igual que en el modelo logit, ambas variables son estadísticamente significativas con p-valores cercanos a 0.

Para comparar ambos modelos, podemos utilizar criterios como el AIC (Criterio de Información de Akaike) y el BIC (Criterio de Información Bayesiano). Estos criterios penalizan la complejidad del modelo y ayudan a seleccionar el modelo que mejor se ajusta a los datos sin sobreajustar.

```r
comp <- data.frame(
    Model = c("logit", "probit"),
    AIC = c(AIC(m_logit), AIC(m_probit)),
    BIC = c(BIC(m_logit), BIC(m_probit))
)
print(comp)
```

```
##    Model      AIC      BIC
## 1  logit 27.06147 25.83406
## 2 probit 26.12412 24.89670
```

Conclusión: Ambos modelos capturan un efecto positivo y significativo de 'ronq' sobre la probabilidad de enfermedad cardíaca. No obstante, el modelo *probit* presenta mejor ajuste global: AIC = 26.12 y BIC = 24.90 frente a AIC = 27.06 y BIC = 25.83 del *logit*. Las conclusiones sustantivas sobre el efecto de los ronquidos son equivalentes en ambos modelos.

Ejercicio 2:

Suponga el siguiente experimento: se escoge al azar una moneda (A o B) y luego tiramos un volado con con esa moneda; los resultados posibles son Sol y Águila. El resultado de este experimento es:

Ejercicio	Moneda	Resultado
1	A	Sol
2	B	Águila
3	A	Águila
4	B	Sol
5	B	Águila
6	Dato faltante	Águila

Si parametrizamos de la siguiente manera:

- θ = Probabilidad que la moneda sea A
- θ_A = Probabilidad que el resultado sea Sol dado que es la moneda A
- θ_B = Probabilidad que el resultado sea Sol dado que es la moneda B

Utilice las técnicas del algoritmo EM para estimar los parámetros θ, θ_A y θ_B tomando en cuenta la presencia de datos faltantes.

Solución:

Primero codificamos los datos (moneda observada o faltante y resultado del volado).

```
moneda <- c('A','B','A','B','B', NA)
resultado <- c('Sol','Aguila','Aguila','Sol','Aguila','Aguila')
df_em <- data.frame(moneda, resultado, stringsAsFactors = FALSE)
df_em
```

```
##   moneda resultado
## 1      A       Sol
## 2      B    Aguila
## 3      A    Aguila
## 4      B       Sol
## 5      B    Aguila
## 6   <NA>    Aguila
```

Implementamos el algoritmo EM. En la E, calculamos la probabilidad posterior de que la observación faltante provenga de la moneda A dado su resultado; en la M, actualizamos θ, θ_A y θ_B con los conteos esperados.

```
em_moneda <- function(df, theta=0.5, thetaA=0.5, thetaB=0.5, tol=1e-10, maxit=1000){
    # log-verosimilitud observada
```

```
    loglik <- function(th, thA, thB){
        sapply(seq_len(nrow(df)), function(i){
            res <- df$resultado[i]
            if (is.na(df$moneda[i])){
                pA <- th * ifelse(res=='Sol', thA, 1-thA)
                pB <- (1-th) * ifelse(res=='Sol', thB, 1-thB)
                log(pA + pB)
            } else if (df$moneda[i]=='A'){
                log(th) + log(ifelse(res=='Sol', thA, 1-thA))
            } else {
                log(1-th) + log(ifelse(res=='Sol', thB, 1-thB))
            }
        }) |> sum()
    }

    it <- 0L; conv <- FALSE
    repeat {
        it <- it + 1L
        # E-step: pesos gamma para las observaciones con moneda faltante
        res_miss <- df$resultado[is.na(df$moneda)]
        if (length(res_miss) == 0L) gammaA <- numeric(0) else {
            pA_miss <- theta * ifelse(res_miss=='Sol', thetaA, 1-thetaA)
            pB_miss <- (1-theta) * ifelse(res_miss=='Sol', thetaB, 1-thetaB)
            gammaA <- pA_miss / (pA_miss + pB_miss)  # P(moneda=A | resultado)
        }

        # Conteos esperados
        nA_k <- sum(df$moneda=='A', na.rm=TRUE)
        nB_k <- sum(df$moneda=='B', na.rm=TRUE)
        n_miss <- sum(is.na(df$moneda))
        nA_exp <- nA_k + sum(gammaA)
        nB_exp <- nB_k + n_miss - sum(gammaA)

        solA_k <- sum(df$moneda=='A' & df$resultado=='Sol', na.rm=TRUE)
        solB_k <- sum(df$moneda=='B' & df$resultado=='Sol', na.rm=TRUE)
        sol_miss <- (res_miss == 'Sol')
        solA_exp <- solA_k + sum(gammaA[sol_miss])
        solB_exp <- solB_k + sum((1-gammaA)[sol_miss])

        # M-step
        theta_new  <- nA_exp / (nA_exp + nB_exp)
        thetaA_new <- ifelse(nA_exp>0, solA_exp / nA_exp, thetaA)
        thetaB_new <- ifelse(nB_exp>0, solB_exp / nB_exp, thetaB)

        # Criterio de convergencia
        delta <- max(abs(c(theta_new-theta, thetaA_new-thetaA, thetaB_new-thetaB)))
        theta <- theta_new; thetaA <- thetaA_new; thetaB <- thetaB_new
```

```
        if (delta < tol || it >= maxit) { conv <- (delta < tol); break }
    }

    list(theta=theta, thetaA=thetaA, thetaB=thetaB, iter=it, converged=conv,
         logLik=loglik(theta, thetaA, thetaB),
         gamma_missing=if (exists('gammaA')) gammaA else NA_real_)
}
```

Ahora, ajustamos el modelo EM a los datos, utilizando como valores iniciales $\theta = 0.6$, $\theta_A = 0.6$ y $\theta_B = 0.5$. Aunque pueden usarse otros valores.

```
set.seed(1) # Para reproducibilidad
fit_em <- em_moneda(df_em, theta=0.5, thetaA=0.6, thetaB=0.5)
fit_em

## $theta
## [1] 0.3888889
##
## $thetaA
## [1] 0.4285714
##
## $thetaB
## [1] 0.2727273
##
## $iter
## [1] 16
##
## $converged
## [1] TRUE
##
## $logLik
## [1] -7.114922
##
## $gamma_missing
## [1] 0.3333333
```

De esta forma, obtenemos los valores estimados de los parámetros.

$$\theta \approx 0.39, \quad \theta_A \approx 0.43, \quad \theta_B \approx 0.27$$

Con las estimaciones finales también podemos calcular la probabilidad posterior de que la observación faltante (resultado = Águila) corresponda a la moneda A.

```
theta  <- fit_em$theta
thetaA <- fit_em$thetaA
thetaB <- fit_em$thetaB
```

```r
# Posterior de A para la observación faltante con resultado "Aguila"
gamma_A <- theta*(1-thetaA) / (theta*(1-thetaA) + (1-theta)*(1-thetaB))
gamma_A #£

## [1] 0.3333333
```

Conclusión: El algoritmo EM converge rápidamente. Las estimaciones típicas (con los datos dados) son $\hat{\theta} \approx 0.39$, $\hat{\theta}_A \approx 0.43$ y $\hat{\theta}_B \approx 0.27$. Esto sugiere que la moneda A se selecciona alrededor del 39 % de las veces; la probabilidad de *Sol* es mayor con la moneda A que con la B. Para la observación con moneda desconocida y resultado *Águila*, la probabilidad posterior de que se haya usado la moneda A es ≈ 0.34, por lo que es más probable que proviniera de la moneda B.

Ejercicio 3:

Ratones bajo tratamiento o no para prolongar su supervivencia después de una cirugía invasiva.

Datos:

Tratamiento: 94, 197, 16, 38, 99, 141, 23.

Control: 52, 104, 146, 10, 51, 30, 40, 27, 46.

1. Utilice técnicas de Bootstrap para determinar si existe diferencia significativa entre las medias.

2. Compare los resultados del inciso 1 con una prueba t estándar para comparación de medias.

3. Utilice técnicas de Bootstrap para determinar si existe diferencia significativa entre las medianas.

Solución:

Primero ingresamos los datos de supervivencia para ambos grupos.

```
trat <- c(94, 197, 16, 38, 99, 141, 23)
ctrl <- c(52, 104, 146, 10, 51, 30, 40, 27, 46)
```

Calculamos estadísticas descriptivas básicas (tamaño, media y mediana) para tener una referencia inicial.

```
data.frame(
    grupo = c("Tratamiento", "Control"),
    n = c(length(trat), length(ctrl)),
    media = c(mean(trat), mean(ctrl)),
    mediana = c(median(trat), median(ctrl))
)
```

```
##         grupo n    media mediana
## 1 Tratamiento 7 86.85714      94
## 2     Control 9 56.22222      46
```

Definimos una función genérica de bootstrap para la diferencia entre grupos en media o mediana. En cada réplica re-muestreamos con reemplazo de cada grupo por separado y computamos la estadística.

```
boot_2samp <- function(x, y, B = 20000L, stat = c("mean", "median")){
    stat <- match.arg(stat)
    sfun <- if (stat == "mean") mean else median
```

```
    n1 <- length(x); n2 <- length(y)
    obs <- sfun(x) - sfun(y)
    boots <- replicate(B, {
        bx <- sample(x, n1, replace = TRUE)
        by <- sample(y, n2, replace = TRUE)
        sfun(bx) - sfun(by)
    })
    list(obs = obs, boots = boots)
}
```

3.1.1 Diferencia significativa entre medias

Aplicamos bootstrap para la diferencia de medias. Fijamos semilla para reproducibilidad.

```
set.seed(2025)
res_mean <- boot_2samp(trat, ctrl, B = 20000, stat = "mean")
```

Obtenemos un intervalo de confianza al 95 % por percentiles y un p-valor a dos colas centrando la distribución bootstrap en su media. Planteamos la prueba de hipótesis para la diferencia de medias con bootstrap: $H_0 : \mu_{trat} - \mu_{ctrl} = 0$ frente a $H_1 : \mu_{trat} - \mu_{ctrl} \neq 0$ al nivel $\alpha = 0.05$. Tomamos la decisión con el p-valor bootstrap calculado.

```
ci_mean <- quantile(res_mean$boots, c(0.025, 0.975))
center_m <- mean(res_mean$boots)
p_mean <- mean(abs(res_mean$boots - center_m) >= abs(res_mean$obs - center_m))
list(
    diferencia_media_obs = res_mean$obs,
    ci_media_95 = unname(ci_mean),
    p_boot_media_2colas = p_mean
) #£
```

```
## $diferencia_media_obs
## [1] 30.63492
##
## $ci_media_95
## [1] -20.36508  85.34960
##
## $p_boot_media_2colas
## [1] 0.98365
```

Conclusión: La diferencia media observada fue de 30.63 unidades a favor del grupo con tratamiento. El p-valor bootstrap fue de 0.98, muy superior a 0.05, por lo que no rechazamos H_0 y concluimos que no hay evidencia suficiente para afirmar que la diferencia de medias es significativa. Consistentemente, el IC del 95 % por percentiles contiene 0, indicando que la diferencia observada es compatible con la variabilidad muestral.

3.1.2 Prueba t estándar para diferencia de medias

Comparamos con una prueba t de Welch (dos muestras, varianzas desiguales) para la diferencia de medias.

```
tt_welch <- t.test(trat, ctrl, var.equal = FALSE, alternative = "two.sided")
tt_welch


##
##  Welch Two Sample t-test
##
## data:  trat and ctrl
## t = 1.0587, df = 9.6545, p-value = 0.3155
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -34.15279  95.42263
## sample estimates:
## mean of x mean of y
##  86.85714  56.22222
```

En esta ocasión, el p-valor de la prueba fue de 0.3155, por lo que no se rechaza H_0 la igualdad de medias. El IC del 95 % para la diferencia de medias fue $(-34.15, 95.42)$, que nuevamente incluye al 0 y es muy amplio, señal de alta incertidumbre con el tamaño muestral y variablidad presentes.

Conclusión: La prueba t de Welch coincide con el bootstrap en no rechazar H_0 y no encontrar evidencia concluyente de diferencia significativa entre las medias de ambos grupos. El IC del 95 % es amplio y contiene 0, indicando que la diferencia observada puede ser atribuible a la variabilidad muestral.

3.1.3 Diferencia significativa entre medianas

Ahora repetimos el procedimiento de bootstrap para la diferencia de medianas.

```
set.seed(2025)
res_med <- boot_2samp(trat, ctrl, B = 20000, stat = "median")
```

Calculamos el intervalo de confianza al 95 % por percentiles y un p-valor bootstrap a dos colas para la mediana.

```
ci_med <- quantile(res_med$boots, c(0.025, 0.975))
center_md <- mean(res_med$boots)
p_med <- mean(abs(res_med$boots - center_md) >= abs(res_med$obs - center_md))
list(
    diferencia_mediana_obs = res_med$obs,
    ci_mediana_95 = unname(ci_med),
    p_boot_mediana_2colas = p_med
) #£
```

```
## $diferencia_mediana_obs
## [1] 48
##
## $ci_mediana_95
## [1] -29 111
##
## $p_boot_mediana_2colas
## [1] 0.8659
```

La diferencia entre las medianas observada fue de 48 unidades a favor del grupo con tratamiento. El p-valor bootstrap fue de 0.86, muy superior a 0.05, por lo que no rechazamos H_0 de que las medianas son iguales. El IC del 95 % por percentiles fue $(-29, 111)$, que nuevamente contiene al 0, indicando que la diferencia observada es compatible con la variabilidad muestral.