De los ejemplos estudiados en el capitulo 2 (capitulo 6):

x: vector no vacío de valores numéricos (la muestra).

alternative: tipo de prueba de hipótesis. Los posibles valores son “two.sided” (prueba bilateral),

“greater” (hipótesis unilateral para suponer que la estimación puntual es mayor que el valor nulo) o

“less” (hipótesis unilateral para suponer que la estimación puntual es menor que el valor nulo).

mu: valor nulo.

conf.level: nivel de confianza.

inferencia con la media de una muestra usando la distribución t.

1 library ( ggpubr )

2

3 # Cargar los datos y guardarlos como data frame .

4 tiempo <- c(411.5538, 393.2753, 445.8905, 411.4022, 498.8969,

5 388.6731, 430.0382, 469.4734, 409.5844, 442.0800,

6 418.1169, 408.4110, 463.3733,407.0908, 516.5222)

7 dataframe <- data . frame ( tiempo )

8

9 # Verificar si la distribuci ón se acerca a la normal .

10 ggqqplot ( data = dataframe ,

11 x = " tiempo ",

12 color = " steelblue ",

13 xlab = "Teó rico ",

14 ylab = " Muestra ",

15 title = "Grá fico Q-Q muestra v/s distr . normal ")

16

17 # Calcular estimadores puntuales y estad í sticos ú tiles .

18 mu\_0 = 500

19 alfa <- 0.025

20

21 # Aplicar la prueba t de Student .

22 t. test (tiempo ,

23 alternative = " less ",

24 mu = mu\_0,

25 conf . level = 1 - alfa )

El valor para el puntaje T es t = −6, 6801.

La distribución t tiene df = 14 grados de libertad.

El valor p obtenido es p = 5, 21 ・ 10−6.

El intervalo de confianza obtenido es (−∞; 441, 103].

La media de la muestra es x = 434, 2921.

Un equipo médico desea determinar si una nueva vacuna A es más efectiva que otra vacuna B, a fin de

inmunizar a la población mundial contra una terrible enfermedad. Para ello, se reclutó a un grupo de 28

voluntarios en diferentes países, 15 de los cuales (seleccionados al azar) recibieron la vacuna A y los 13 restantes,la vacuna B. Las concentraciones de anticuerpos (en microgramos por cada mililitro) al cabo de un

mes, para cada voluntario que recibió la vacuna A fueron: 9,534639; 4,458274; 9,961477; 7,831859; 8,587947;4,233574; 0,957678; 10,503414; 10,009914; 23,643719; 4,793084; 13,441451; 7,042911; 5,303227; 9,557578. Asimismo,para quienes recibieron la vacuna B fueron: 16,178461; 15,687254; 4,879188; 13,768377; 15,591773;4,493893; 17,849865; 10,549353; 29,092156; 7,951576; 15,851443; 9,418612; 18,309343.

Las hipótesis a formular en este caso son:

H0: No hay diferencia entre la efectividad promedio de ambas vacunas.

HA: La vacuna A es, en promedio, más efectiva que la B.

1 library ( ggpubr )

3 # Cargar los datos y guardarlos como data frame .

4 instancia <- seq(1, 35, 1)

6 c\_A <- c(9.534639, 4.458274, 9.961477, 7.831859, 8.587947,

7 4.233574, 0.957678, 10.503414, 10.009914, 23.643719,

8 4.793084, 13.441451, 7.042911, 5.303227, 9.557578)

10 data \_A <- data . frame (c\_A)

12 c\_B <- c(16.178461, 15.687254, 4.879188, 13.768377, 15.591773,

13 4.493893, 17.849865, 10.549353, 29.092156, 7.951576,

14 15.851443, 9.418612, 18.309343)

16 data \_B <- data . frame (c\_B)

19 # Verificar si la primera muestra se distribuye de manera cercana

20 # a la normal .

21 ggqqplot ( data = data \_A,

22 x = "c\_A",

23 color = " darkgreen ",

24 xlab = "Teó rico ",

25 ylab = " Muestra ",

26 title = "Grá fico Q-Q muestra A v/s distr . normal ")

28 # Verificar si la primera muestra se distribuye de manera cercana

29 # a la normal .

30 ggqqplot ( data = data \_B,

31 x = "c\_B",

32 color = " red",

33 xlab = "Teó rico ",

34 ylab = " Muestra ",

35 title = "Grá fico Q-Q muestra B v/s distr . normal ")

37 # # Calcular estimadores puntuales y estad í sticos ú tiles .

38 mu\_0 = 0 # En este caso , representa la diferencia entre las medias .

39 alfa <- 0.01

41 # Aplicar la prueba t de Student para dos muestras independientes .

42 resultado <- t. test (x = c\_A,

43 y = c\_B,

44 paired = FALSE ,

45 alternative = " greater ",

46 mu = 0,

47 conf . level = 1 - alfa )

49 resultado:

51 # Calcular la diferencia entre las medias .

52 media \_A <- resultado $ estimate [1]

53 media \_B <- resultado $ estimate [2]

54 media \_A - media \_B

Del capitulo 3 (capitulo 7-8) PODER

Considere que un ingeniero en informática cuenta con dos programas, A y B, que resuelven instancias de

tamaño fijo de un mismo problema. Suponga que se sabe que el tiempo de ejecución sigue una distribución

normal con desviación estándar \_ = 12 milisegundos, y que el ingeniero desea probar si existe diferencia entre los tiempos de ejecución de ambos algoritmos, con un nivel de significación \_ = 0; 05 usando una muestra de 36 observaciones independientes. En este caso, las hipótesis correspondientes son:

H0: \_(Ai􀀀Bi) = 0, es decir que la media de las diferencias en el tiempo de ejecución necesitado por los

algoritmos A y B, para cada posible instancia i, es cero

HA: \_(Ai􀀀Bi) 6= 0

pwr.t.test(n, d, sig.level, power, type, alternative)incluida en el paquete pwr, donde:

n: tamaño de la muestra (por cada grupo, si corresponde).

d: tamaño del efecto (d de Cohen).

sig.level: nivel de significación.

power: poder de la prueba.

type: tipo de prueba t de Student (“two.sample” para diferencia de medias, “one.sample” para

una sola muestra o “paired” para dos muestras pareadas).

alternative: tipo de hipótesis alternativa (“greater” o “less” si es unilateral, “two.sided”

si es bilateral).

1 library ( ggpubr )

# Par á metros conocidos .

4 sigma <- 12

5 alfa <- 0.05

6 n <- 36

# Cá lculo del error está ndar .

9 SE <- sigma / sqrt (n)

11 # Grá fico de la distribuci ón de la diferencia de medias si la diferencia

# fuera nula , con regiones de rechazo de la hipó tesis nula .

13 media \_ nula <- 0

14 Z\_ critico <- qnorm ( alfa /2, mean = media \_nula , sd = SE , lower . tail = FALSE )

15 q\_ critico \_ inferior <- media \_ nula - Z\_ critico

16 q\_ critico \_ superior <- media \_ nula + Z\_ critico

17

18 x <- seq(-6 \* SE , 4 \* SE , 0.01)

19 y <- dnorm (x, mean = media \_nula , sd = SE)

20 df <- data . frame (x, y)

21

22 g <- ggplot ( data = df , aes(x))

23

24 g <- g + stat \_ function (

25 fun = dnorm ,

26 args = list ( mean = media \_nula , sd = SE),

27 colour = "red ", size = 1)

28

29 g <- g + ylab ("")

30 g <- g + scale \_y\_ continuous ( breaks = NULL )

31 g <- g + scale \_x\_ continuous ( name = " Diferencia en tiempos de ejecuci ón [ms]",

32 breaks = seq (-6, 4, 2))

33

34 g <- g + theme \_ pubr ()

35

36 g <- g + geom \_ area ( data = subset (df , x < q\_ critico \_ inferior ),

37 aes (y = y),

38 colour = "red",

39 fill = "red",

40 alpha = 0.5)

41

42 g <- g + geom \_ area ( data = subset (df , x > q\_ critico \_ superior ),

43 aes (y = y),

44 colour = "red ",

45 fill = "red",

46 alpha = 0.5)

47

48 print (g) # Figura 7.5

49

50 # Grá fico de la distribuci ón de la diferencia de medias si la diferencia

51 # fuera -4.

52 media \_ efecto <- -4

53

54 x1 <- seq(-6 \* SE , 4 \* SE , 0.01)

55 y1 <- dnorm (x, mean = media \_efecto , sd = SE)

56 df1 <- data . frame (x1, y1)

57

58 g <- g + stat \_ function (

59 fun = dnorm ,

60 args = list ( mean = media \_efecto , sd = SE),

61 colour = " blue ", size = 1)

62

63 g <- g + ylab ("")

64

65 g <- g + geom \_ area ( data = subset (df1, x < q\_ critico \_ inferior ),

66 aes (x = x1, y = y1),

67 colour = " blue ",

68 fill = " blue ",

69 alpha = 0.5)

g <- g + geom \_ area ( data = subset (df1, x > q\_ critico \_ superior ),

72 aes (x = x1, y = y1),

73 colour = " blue ",

74 fill = " blue ",

75 alpha = 0.5)

77 print (g) # Figura 7.6

79 # Cá lculo del poder de acuerdo al aná lisis teó rico .

80 poder <- pnorm (q\_ critico \_ inferior ,

81 mean = media \_efecto ,

82 sd = SE ,

83 lower . tail = TRUE )

84 + pnorm (q\_ critico \_ superior ,

85 mean = media \_efecto ,

86 sd = SE ,

87 lower . tail = FALSE )

89 print ( poder )

91 # Cá lculo del poder usando la funci ón power .t. test ().

92 poder \_1 <- power .t. test (n=36,

93 delta =4,

94 sd=12,

95 sig . level = .05,

96 type =" paired ",

97 alternative = " two. sided ")

99 print ( poder \_1)

101 # Cá lculo del poder usando la funci ón pwr.t. test ().

102 library (pwr)

104 d\_ cohen <- (4 / sigma ) \* ((n - 2) / (n - 1.25))

106 poder \_2 <- pwr.t. test (n = 36,

107 sig . level = alfa ,

108 type = " paired ",

109 alternative = " two. sided ",

110 d = d\_ cohen )

112 print ( poder \_2)

El valor p asociado, calculado en R mediante la llamada a la función pnorm(-1,6036, lower.tail=FALSE),

obteniéndose como resultado p = 0; 9456. En consecuencia, la evidencia no es suficiente para rechazar la hipótesis

nula, por lo que se concluye, con 95% de confianza, que no es cierto que el algoritmo se ejecute en

menos de 25 segundos para más del 70 %.

La prueba de hipótesis anterior puede efectuarse en R como se muestra en el script 8.1, usando para ello la

función prop.test(), cuyos principales parámetros son:

x: cantidad de éxitos en la muestra.

n: tamaño de la muestra.

p: valor nulo (por defecto, p=0).

alternative: tipo de hipótesis alternativa, por defecto bilateral (alternative=“two.sided”), y

valores “less” y “greater” para hipótesis unilaterales.

conf.level: nivel de confianza (conf.level=0.95 por defecto).

Script 8.1: prueba de hipótesis para una proporción.

1 # Tama ño, proporci ón de é xito y cantidad de é xitos de la muestra .

2 n <- 150

3 p\_ muestra <- 0.64

4 exitos <- p\_ muestra \* n

# Valor nulo y nivel de significaci ón.

7 p0 <- 0.7

8 alfa <- 0.05

10 # Prueba de hipó tesis .

11 prueba <- prop . test ( exitos ,

12 n = n,

13 p = p0,

14 alternative = " greater ",

15 conf . level = 1-alfa )

17 print ( prueba )

Script 8.2: prueba de hipótesis para diferencia igual a 0 entre dos proporciones.

1 # Cantidad de é xitos ( hombres , mujeres ).

2 exitos <- c(26, 20)

34

# Tama ño de la muestra ( hombres , mujeres ).

5 n <- c(48, 42)

67

# Nivel de significaci ón.

8 alfa <- 0.05

9

10 # Prueba de la hipó tesis .

11 prueba <- prop . test ( exitos ,

12 n = n,

13 alternative = " two. sided ",

14 conf . level = 1-alfa )

15

16 print ( prueba )

prueba de hipótesis para diferencia distinta de 0 entre dos proporciones.

1 # Tama ños de las muestras .

2 # 1: hombres

3 # 2: mujeres

4 n1 <- 89

5 n2 <- 61

# Cantidad de é xitos en las muestras .

8 exitos 1 <- 45

9 exitos 2 <- 21

11 # Valor nulo

12 nulo <- 0.01

14 # Probabilidades de é xito estimadas .

15 p\_est1 <- exitos 1/n1

16 p\_est2 <- exitos 2/n2

18 # Diferencia de las proporciones .

19 dif <- p\_est1 - p\_est2

21 # Error está ndar .

22 err <- sqrt (p\_est1 \* (1 - p\_est 1) / n1 + p\_est 2 \* (1 - p\_est2) / n2)

24 # Estad í stico de prueba .

25 Z <- (dif - nulo )/err

27 # Valor p.

28 p <- pnorm (Z, lower . tail = FALSE )

29 print (p)

DEL CAPITULO 9

PRUEBA DE FISHER

Considerando un nivel de significación \_ = 0; 05, se falla al rechazar la hipótesis nula. En consecuencia, se

concluye que, con 95% de confianza, no hay una asociación estadísticamente significativa entre la cantidad

de infectados y la vacuna recibida.

En R, esta prueba puede efectuarse mediante la función fisher.test(x, conf.level), donde x corresponde

a la tabla de contingencia y conf.level, al nivel de confianza. El script 9.1 muestra el resultado

para el ejemplo.

Script 9.1: prueba de Fisher.

1 # Construir la tabla de contingencia

2 vacuna <- c(rep (" Pfazer ", 6), rep (" Simovak ", 11))

3 resultado <- c(rep(" Sano ", 12), rep(" Infectado ", 5))

4 datos <- data . frame ( resultado , vacuna )

5 tabla <- xtabs (~. , datos )

# Aplicar prueba exacta de Fisher

8 alfa <- 0.05

9 prueba <- fisher . test (tabla , 1-alfa )

10 print ( prueba )

EJEMPLO PRUEBA DE MCNEMAR

prueba de McNemar.

1 # Construir la tabla de contingencia

2 alumno <- seq (1:25)

3 modelo \_1 <- c(rep(" Correcto ", 16), rep(" Incorrecto ", 9))

4 modelo \_2 <- c(rep(" Correcto ", 9), rep(" Incorrecto ", 11), rep(" Correcto ", 5))

5 datos <- data . frame (alumno , modelo \_2, modelo \_1)

6 tabla <- table ( modelo \_2, modelo \_1)

7 addmargins ( tabla )

89

# Aplicar prueba de McNemar

10 mcnemar . test ( tabla )

PRUEBAS CHI-CUADRADO

HOMOGENEIDAD

En R, la prueba chi-cuadrado de homogeneidad puede realizase, como muestra el script 9.3, usando la función

chisq.test(x, p), donde:

x: vector con la cantidad de observaciones para cada grupo.

p: vector con la probabilidad esperada para cada grupo.

Tenga en cuenta que el valor p obtenido usando R varía ligeramente, pues todos los valores en este desarrollo

han sido redondeados.

Script 9.3: prueba chi-cuadrado de homogeneidad.

1 # Construir la matriz de datos

2 programadores <- c(42, 56, 51, 27, 24)

3 programadoras <- c(25, 24, 27, 15, 9)

tabla <- as. table ( rbind ( programadores , programadoras ))

6 dimnames ( tabla ) <- list ( sexo = c(" programadores ", " programadoras "),

7 lenguajes = c("C", " Java ", " Python ", " Ruby ", " Otro "))

# Hacer la prueba chi - cuadrado de homogeneidad

10 prueba <- chisq . test (tabla , correct = FALSE )

11 print ( prueba )

BONDAD DE AJUSTE

H0: las proporciones de especialistas en cada lenguaje son las mismas para la nómina y la muestra.

HA: las proporciones de especialistas en cada lenguaje son diferentes en la nómina que en la muestra.

En este caso, se puede proceder de igual manera que para la prueba de bondad de ajuste, como muestra el

script 9.4. Para este ejemplo, el valor p resultante es p = 0; 823, por lo que se falla al rechazar la hipótesis

nula con niveles de significación típicos (como \_ = 0; 05 o \_ = 0; 01. En consecuencia, se concluye con 95%

de confianza que la muestra seleccionada es, en efecto, representativa de la nómina de programadores de la

empresa.

Script 9.4: prueba chi-cuadrado de bondad de ajuste.

1 # Construir la matriz de datos

2 nomina <- c(43, 13, 34, 9, 11)

3 muestra <- c(7, 2, 7, 3, 1)

tabla <- as. table ( rbind (nomina , muestra ))

6 dimnames ( tabla ) <- list ( sexo = c("Nó mina ", " Muestra "),

7 lenguajes = c("C", " Java ", " Python ", " Ruby ", " Otro "))

# Hacer la prueba chi - cuadrado

prueba <- chisq . test (tabla , correct = FALSE )

11 print ( prueba )

INDEPENDENCIA

H0: las variables clase y forma del sombrero son independientes.

HA: las variables clase y forma del sombrero son dependientes.

Al ejecutar la prueba en R (script 9.5) se obtiene que el valor para el estadístico de prueba es \_2 = 485; 64,

con \_ = 4 grados de libertad y un p-valor < 2; 2 \_ 10􀀀16.

Script 9.5: prueba chi-cuadrado de independencia.

1 # Crear tabla de contingencia ( tabla 9.4).

2 tabla <- cbind (c(404, 48), c(1948, 1708), c(32, 0), c(228, 600), c(1596, 1556))

3 colnames ( tabla ) <- c(" campana ", " convexo ", " hundido ", " nudoso ", " plano ")

4 rownames ( tabla ) <- c(" comestible ", " venenoso ")

5 tabla <- as. table ( tabla )

# # Prueba chi - cuadrado de independencia .

8 prueba <- chisq . test (tabla , correct = FALSE )

9 print ( prueba )

EJEMPLOS CAPITULO 10 : ANOVA

procedimiento ANOVA de una vía para muestras independientes.

1 library ( ggpubr )

2 library (ez)

# Crear el data frame .

5 a <- c(23, 19, 25, 23, 20)

6 b <- c(26, 24, 28, 23, 29)

7 c <- c(19, 24, 20, 21, 17)

Tiempo <- c(a, b, c)

10 Algoritmo <- c(rep("A", length (a)),

11 rep ("B", length (b)),

12 rep ("C", length (c)))

14 Algoritmo <- factor ( Algoritmo )

15 instancia <- factor (seq(1, 15, by = 1))

16 datos <- data . frame ( instancia , Algoritmo , Tiempo )

18 # Comprobci ón de normalidad .

19 g <- ggqqplot (datos ,

20 x = " Tiempo ",

21 y = " Algoritmo ",

22 color = " Algoritmo ")

g <- g + facet \_ wrap (~ Algoritmo )

25 g <- g + rremove ("x. ticks ") + rremove ("x. text ")

26 g <- g + rremove ("y. ticks ") + rremove ("y. text ")

27 g <- g + rremove (" axis . title ")

28 print (g) # Figura 11.1

30 # Procedimiento ANOVA con aov ().

31 prueba <- aov ( Tiempo ~ Algoritmo , data = datos )

32 print ( summary ( prueba ))

34 # Procedimiento ANOVA con ezANOVA ().

35 prueba 2 <- ezANOVA (

36 data = datos ,

37 dv = Tiempo ,

38 between = Algoritmo ,

39 wid = instancia ,

40 return \_aov = TRUE )

42 print ( prueba 2)

44 # Grá fico del tama ño del efecto .

45 g2 <- ezPlot (

46 data = datos ,

47 dv = Tiempo ,

48 wid = instancia ,

49 between = Algoritmo ,

50 y\_lab = " Tiempo promedio de ejecuci ón [ms]",

51 x = Algoritmo

52 )

54 print (g2) # Figura 10.2