Solucion Taller de problemas inferencia 2023 MAT3 GIN

Jesús Castillo Benito

Contenidos

1 Taller INDIVIDUAL Problemas evaluable 22-23: Estadística Inferencial **1**

1.1 Problema 1: Contraste de parámetros de dos muestras. Test AB. (6 puntos) 1

1.2 Problema 2: Bondad de ajuste. La ley de Benford. (4 puntos) 9

1.3 Problema 3: Homegeneidad e independencia. (3 puntos). 13

1.4 Problema 4: Contraste de proporciones de dos muestras independientes. (3. puntos) 14

1.5 Problema 5 : Contraste de proporciones de dos muestras emparejadas. (2. puntos) 14

1 Taller INDIVIDUAL Problemas evaluable 22-23: Estadística Inferencial

Cada apartado es 1 punto. Total 18 puntos

Se trata de resolver los siguientes problemas y cuestiones en un fichero Rmd y su salida en un informe en html, word o pdf o escrito manualmente y escaneado.

1.1 Problema 1: Contraste de parámetros de dos muestras. Test AB. (6 puntos)

Se quiere evaluar dos interfaces gráficas para un vídeo juego la tipo A que es la actual y una nueva tipo B. Se selecciona dos muestras de jugadores independientes la primera prueba la interfaz A y la segunda la B. En cada muestra se observa el tiempo utilizado para completar una fase del juego en minutos. Las muestras son de tamaños $n_A = 1000$ y $n_B = 890$.

Los datos están adjuntos a los enunciados, en la carpeta `datasets` en un ficheros `AB.csv` que contien las variables tiempo y muestra que vale A o B.

1. Cargad de datos y calculad estadísticos descriptivos básicos y diagramas de caja e histogramas muestrales, utilizad la función `density`, comparativos de las dos muestras.
2. Estudiad si podemos aceptar que las muestras son normales con el test K-S-L, Ardenson-Darling test, Shapiro-Wilks y Dagostino-Pearson.
3. Calcular el estadístico de contraste del test K-S-L para la muestra A y comprobad el resultado.
4. Comprobad con el test de Fisher de razón de varianzas si las varianza de las dos muestras son iguales contra que son distintas. Tenéis que resolver el test de Fisher con R y de forma manual y el de Flinger de R y decidir utilizando el p-valor.
5. Con la información anterior elegid el contraste adecuado para saber si hay evidencia de que la la nueva interfaz mejora el tiempo de la actual. Resolver manualmente definiendo adecuadamente las hipótesis y decidid según el p-valor.
6. Calculad e interpretar el intervalo de confianza de los estadísticos del los test de medias y el de Fisher de los apartados 4 y 5.

1.1.1 Solucion 1

Apartado 1

Primero cargamos los datos del fichero AB.csv:

```
AB <- read.csv("datasets/AB.csv")
```

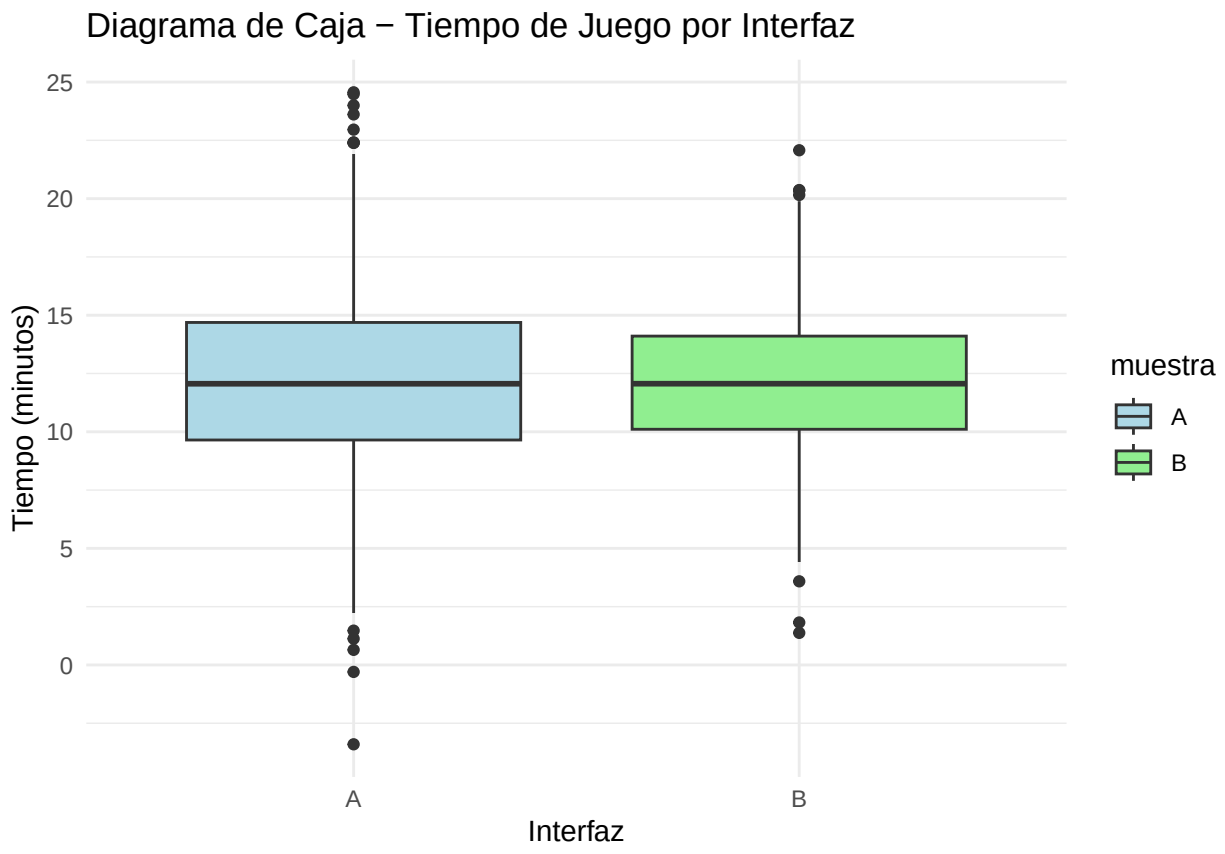
Una vez tenemos los datos cargados, realizamos un cálculo de los estadísticos descriptivos básicos.

```
AB %>% group_by(muestra) %>% summarise(N = n(),
                                       Media_muestra = mean(tiempo),
                                       Desviacion_muestra = sd(tiempo),
                                       Mediana_muestra = median(tiempo),
                                       Max_muestra = max(tiempo),
                                       Min_muestra = min(tiempo))
```

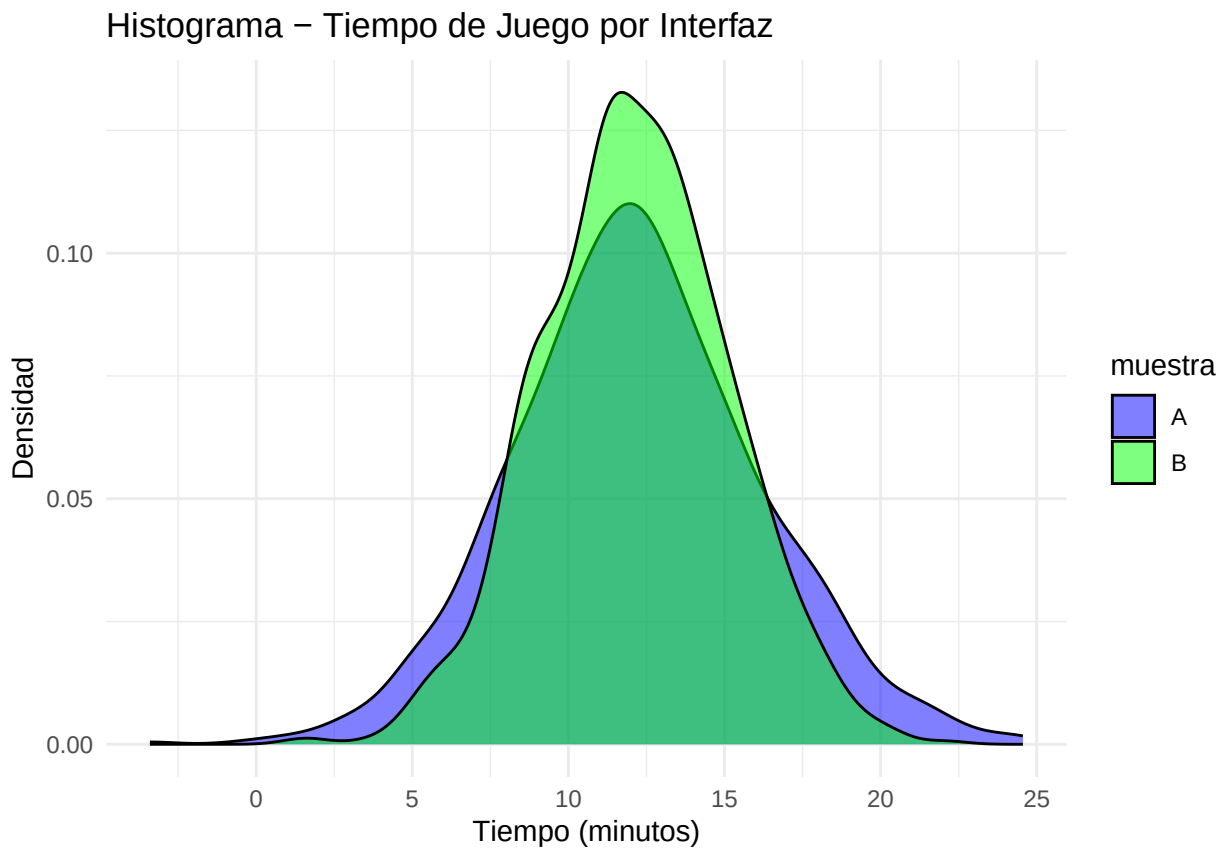
```
## # A tibble: 2 x 7
##   muestra     N Media_muestra Desviacion_muestra Mediana_muestra Max_muestra
##   <chr>   <int>         <dbl>              <dbl>           <dbl>       <dbl>
## 1 A        1000          12.2               3.93            12.1        24.6
## 2 B         890          12.1               2.99            12.1        22.1
## # i 1 more variable: Min_muestra <dbl>
```

Una vez tenemos calculados los estadísticos básicos, podemos realizar el diagrama de cajas y el histograma comparativo de las muestras.

```
ggplot(AB, aes(x = muestra, y = tiempo, fill = muestra)) +
  geom_boxplot() +
  labs(title = "Diagrama de Caja - Tiempo de Juego por Interfaz",
       y = "Tiempo (minutos)", x = "Interfaz") +
  scale_fill_manual(values = c("lightblue", "lightgreen")) +
  theme_minimal()
```




```
ggplot(AB, aes(x = tiempo, fill = muestra)) +
  geom_density(alpha = 0.5, position = "identity") +
  labs(title = "Histograma - Tiempo de Juego por Interfaz",
       y = "Densidad", x = "Tiempo (minutos)") +
  scale_fill_manual(values = c("blue", "green")) +
  theme_minimal()
```



Apartado 2

Utilizaremos las bibliotecas nortest y moments para realizar los tests sobre la muestra de datos. Primero cargamos los datos del fichero AB.csv y dividimos los datos en dos grupos según el tipo de interfaz.

```
datos <- read.csv("datasets/AB.csv")

datos_A <- datos$tiempo[datos$muestra == "A"]
datos_B <- datos$tiempo[datos$muestra == "B"]
```

Realizamos las pruebas de normalidad en cada grupo de datos.

```
# Test de Kolmogorov-Smirnov-Lilliefors
lillie.test(datos_A)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  datos_A
## D = 0.026624, p-value = 0.09182
```

```
lillie.test(datos_B)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
```

```
##
## data:  datos_B
## D = 0.016554, p-value = 0.8021
```

```
# Test de Anderson-Darling
ad.test(datos_A)
```

```
##
##  Anderson-Darling normality test
##
## data:  datos_A
## A = 0.74028, p-value = 0.05364
```

```
ad.test(datos_B)
```

```
##
##  Anderson-Darling normality test
##
## data:  datos_B
## A = 0.18654, p-value = 0.9048
```

```
# Test de Shapiro-Wilks
shapiro.test(datos_A)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_A
## W = 0.99759, p-value = 0.1496
```

```
shapiro.test(datos_B)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_B
## W = 0.99905, p-value = 0.9375
```

```
# Test de D'Agostino-Pearson
agostino.test(datos_A)
```

```
##
##  D'Agostino skewness test
##
## data:  datos_A
## skew = 0.069448, z = 0.902119, p-value = 0.367
## alternative hypothesis: data have a skewness
```

```r
agostino.test(datos_B)
```

```
##
##  D'Agostino skewness test
##
## data:  datos_B
## skew = -0.018649, z = -0.228993, p-value = 0.8189
## alternative hypothesis: data have a skewness
```

Con los datos obtenidos de las pruebas de normalidad, podemos observar que no podemos rechazar la hipótesis nula de que los datos siguen una distribución normal para ambas muestras. Esto se debe a que todos los valores de p son mayores al nivel de significancia de 0.05. Por tanto, podemos asumir que los datos siguen una distribución normal.

Apartado 3

Realizamos el test de Kolmogorov-Smirnov-Lilliefors para la muestra A.

```r
lillie.test(datos_A)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  datos_A
## D = 0.026624, p-value = 0.09182
```

Al realizar el test de Kolmogorov-Smirnov-Lilliefors, obtenemos un valor de p = 0.09182, que es mayor al nivel de significancia de 0.05. Por tanto, no podemos rechazar la hipótesis nula de que los datos siguen una distribución normal.

Apartado 4

Primero, realizamos el test de Fisher Utilizando R.

```r
datos_A <- datos$tiempo[datos$muestra == "A"]
datos_B <- datos$tiempo[datos$muestra == "B"]
var.test(datos_A, datos_B)
```

```
##
##  F test to compare two variances
##
## data:  datos_A and datos_B
## F = 1.7229, num df = 999, denom df = 889, p-value < 0.00000000000000022
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.515729 1.957468
## sample estimates:
## ratio of variances
##           1.722912
```

Ahora, realizamos el test de Fisher de forma manual.

```r
var_A <- var(datos_A)
var_B <- var(datos_B)
F_stat <- var_A / var_B
df1 <- length(datos_A) - 1
df2 <- length(datos_B) - 1
p_value <- pf(F_stat, df1, df2, lower.tail = FALSE)
print(paste("Test de Fisher: F =", F_stat, ", p =", p_value))
```

```
## [1] "Test de Fisher: F = 1.7229115499503 , p = 0.000000000000000008161266338176 81"
```

Al realizar el test de Fisher, obtenemos un valor F de 1.722912 y un valor p menor que 0.05. Por tanto, podemos rechazar la hipótesis nula de que las varianzas de las muestras son iguales.

Realizamos el test de Fligner-Killeen.

```r
fligner.test(list(datos_A, datos_B))
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  list(datos_A, datos_B)
## Fligner-Killeen:med chi-squared = 43.044, df = 1, p-value =
## 0.00000000005353
```

Al realizar el test de Fligner-Killeen, obtenemos un valor de $p < 0.05$. Por tanto, podemos rechazar la hipótesis nula de que las varianzas de las muestras son iguales.

Apartado 5

Para determinar si la nueva interfaz mejora el tiempo respecto a la interfaz antigua, realizamos un test de hipótesis para la media de dos muestras independientes. Para ello, utilizaremos el test t de Student. Aquí podemos ver la resolución usando el test proporcionado por R.

```r
datos_A <- datos$tiempo[datos$muestra == "A"]
datos_B <- datos$tiempo[datos$muestra == "B"]
t.test(datos_A, datos_B, alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  datos_A and datos_B
## t = 0.35429, df = 1845.1, p-value = 0.3616
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.2060502        Inf
## sample estimates:
## mean of x mean of y
##  12.17585  12.11932
```

Ahora, realizamos el test de hipótesis para la media de dos muestras independientes de forma manual.

Calculamos primero la media y la desviación estándar de cada muestra.

```r
media_A <- mean(datos_A)
media_B <- mean(datos_B)

sd_A <- sd(datos_A)
sd_B <- sd(datos_B)
```

Después calculamos la longitud de las muestras, y después el estadístico t y los grados de libertad.

```r
n_A <- length(datos_A)
n_B <- length(datos_B)

t_stat <- (media_A - media_B) / sqrt((sd_A^2/n_A) + (sd_B^2/n_B))

df <- n_A + n_B - 2
```

Por último, calculamos el valor p e imprimimos el resultado.

```r
p_value <- pt(t_stat, df, lower.tail = FALSE)

print(paste("Test t: t =", t_stat, ", p =", p_value))
```

```
## [1] "Test t: t = 0.354285951367649 , p = 0.361582079372982"
```

El test t de Student nos devuelve un valor de $p > 0.05$. Por tanto, podemos concluir que la nueva interfaz no mejora el tiempo respecto a la interfaz antigua.

Apartado 6

Primero calcularemos el intervalo de confianza para la diferencia de medias:

No calcularemos la media, la desviación estándar ni la longitud de cada muestra porque ya las hemos calculado en el apartado anterior. Pasaremos a calcualar el valor crítico de la distribución t y el intervalo de confianza.

```r
t_crit <- qt(0.975, df = n_A + n_B - 2)

IC_media <- c((media_A - media_B) - t_crit * sqrt((sd_A^2/n_A) + (sd_B^2/n_B)),
              (media_A - media_B) + t_crit * sqrt((sd_A^2/n_A) + (sd_B^2/n_B)))
```

Ahora calcularemos el intervalo de confianza para la razón de varianzas:

Primero haremos los cálculos del estadístico F y el valor crítico de la distribución F.

```r
F_stat <- var(datos_A) / var(datos_B)

F_crit <- qf(0.975, df1 = n_A - 1, df2 = n_B - 1)
```

Por último calcularemos el intervalo de confianza.

```r
IC_Fisher <- c(F_stat / F_crit, F_stat * F_crit)
```

Los resultados obtenidos son los siguientes:

```r
print(paste("Intervalo de confianza para la diferencia de medias: [", IC_media[1], ",", IC_media[2], "]
```

```
## [1] "Intervalo de confianza para la diferencia de medias: [ -0.256396890640545 , 0.369454105721623 ]
```

```r
print(paste("Intervalo de confianza para la razón de varianzas: [", IC_Fisher[1], ",", IC_Fisher[2], "]
```

```
## [1] "Intervalo de confianza para la razón de varianzas: [ 1.51572920342242 , 1.95841328533463 ]"
```

Podemos concluir lo siguiente:

- El intervalo de confianza incluye el valor 0, lo que indica que la diferencia entre las medias de las dos muestras no es estadísticamente significativa al nivel de confianza del 95%. No hay sugiciente evidencia para afirmar que la nueva interfaz mejora el tiempo respecto a la interfaz antigua.
- El intervalo de confianza para la razón de medias no incluye el valor 1, lo que indica que la variabilidad en los tiempos para completar una fase del juego es estadísticamente diferente entre las dos interfaces gráficas a nivel de confianza del 95%. En particular, dado que la razón de varianzas es mayor que 1, esto sugiere que la interfaz B tiene una mayor variabilidad en los tiempos de finalización en comparación con la interfaz A.

1.2 Problema 2: Bondad de ajuste. La ley de Benford. (4 puntos)

La ley de Benford es una distribución discreta que siguen las frecuencias de los primero dígitos significativos (de 1 a 9) de algunas series de datos curiosas.

Sea una v.a. X con dominio $D_X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ diremos que sigue una ley de Benford si

$$P(X = x) = \log_{10}\left(1 + \frac{1}{x}\right) \text{ para } x \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}\,.$$

Concretamente las probabilidades son

```
## [1] 0.30103000 0.17609126 0.12493874 0.09691001 0.07918125 0.06694679 0.05799195
## [8] 0.05115252 0.04575749
```

	Díg. 1	Díg. 2	Díg. 3	Díg. 4	Díg. 5	Díg. 6	Díg. 7	Díg. 8	Díg. 9
prob	0.30103	0.1760913	0.1249387	0.09691	0.0791812	0.0669468	0.0579919	0.0511525	0.0457575

En general esta distribución se suele encontrar en tablas de datos de resultados de observaciones de funciones científicas, contabilidades, cocientes de algunas distribuciones ...

1. Contrastar con un test χ^2 si el primer dígito significativo de las cubos de los números naturales del 1 al 1000 sigue esa distribución.
2. Contrastar con un test χ^2 si que el segundo dígito significativo de los cubos los números naturales del 1 al 1000 sigue una uniforme discreta de los diez dígitos del 0 al 9.
3. Calcular manualmente el estadístico y el p-valor del los dos contrates anteriores.
4. Dibujad con `R` para los apartados 1 y 2 los diagramas de frecuencias esperados y observados. Comentad estos gráficos.

1.2.1 Solución 2

Apartado 1

Primero calcularemos las frecuencias observadas y las frecuencias esperadas:

Calculamos los cubos de los números naturales del 1 al 1000 y extraemos el primero dígito de cada uno.

```
cubos <- (1:1000)^3
primeros_digitos <- as.numeric(substr(cubos, 1, 1))
```

Con esto calculamos las frecuencias observadas en los primeros dígitos.

```
frecuencias_observadas <- table(primeros_digitos)
frecuencias_observadas
```

```
## primeros_digitos
##   1   2   3   4   5   6   7   8   9
## 226 159 124 106  94  83  74  71  63
```

Ahora calculamos las frecuencias esperadas en los primeros dígitos según la ley de Benford.

```
probabilidades_benford <- c(0.30103000, 0.17609126, 0.12493874, 0.09691001, 0.07918125, 0.06694679, 0.05
frecuencias_esperadas <- probabilidades_benford * length(cubos)
frecuencias_esperadas
```

```
## [1] 301.03000 176.09126 124.93874  96.91001  79.18125  66.94679  57.99195
## [8]  51.15252  45.75749
```

Por último, realizamos el test χ^2.

```
chisq.test(frecuencias_observadas, p = probabilidades_benford)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  frecuencias_observadas
## X-squared = 46.459, df = 8, p-value = 0.0000001944
```

Apartado 2

No calcularemos los cubos de los números naturales del 1 al 1000 porque ya los hemos calculado en el apartado anterior. Pasaremos a extraer el segundo dígito de cada uno y sus frecuencias observadas.

```
segundos_digitos <- as.numeric(substr(cubos, 2, 2))
frecuencias_observadas2 <- table(segundos_digitos)
frecuencias_observadas2
```

```
## segundos_digitos
##   0   1   2   3   4   5   6   7   8   9
## 115 109 106  98 104  97  91  99  89  90
```

Ahora calculamos las frecuencias esperadas en los segundos dígitos según una distribución uniforme discreta de los diez dígitos del 0 al 9.

```r
probabilidades_uniforme <- rep(1/10, 10)
frecuencias_esperadas2 <- probabilidades_uniforme * length(cubos)
frecuencias_esperadas2
```

```
##  [1] 100 100 100 100 100 100 100 100 100 100
```

Por último, realizamos el test χ^2.

```r
chisq.test(frecuencias_observadas2, p = probabilidades_uniforme)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  frecuencias_observadas2
## X-squared = 6.7495, df = 9, p-value = 0.6632
```

Apartado 3

Primero calcularemos el estadístico de contraste para el apartado 1:

Para calcular el χ^2 utilizaremos la siguiente fórmula: $\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$

```r
chi_cuadrado <- sum((frecuencias_observadas - frecuencias_esperadas)^2 / frecuencias_esperadas)
print(paste("Estadístico X^2: ", chi_cuadrado))
```

```
## [1] "Estadístico X^2:  46.4592477127438"
```

Calculamos los grados de libertad y el p-valor:

```r
grados_de_libertad <- length(frecuencias_observadas) - 1
p_valor <- 1 - pchisq(chi_cuadrado, df = grados_de_libertad)
print(paste("p-valor: ", p_valor))
```

```
## [1] "p-valor:  0.000000194386314489314"
```

Ahora calcularemos el estadístico de contraste para el apartado 2:

Para calcular el χ^2 utilizaremos la siguiente fórmula: $\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$

```r
chi_cuadrado2 <- sum((frecuencias_observadas2 - frecuencias_esperadas2)^2 / frecuencias_esperadas2)
print(paste("Estadístico X^2: ", chi_cuadrado2))
```

```
## [1] "Estadístico X^2:  6.74"
```

Calculamos los grados de libertad y el p-valor:

```r
grados_de_libertad2 <- length(frecuencias_observadas2) - 1
p_valor2 <- 1 - pchisq(chi_cuadrado2, df = grados_de_libertad2)
print(paste("p-valor: ", p_valor2))
```

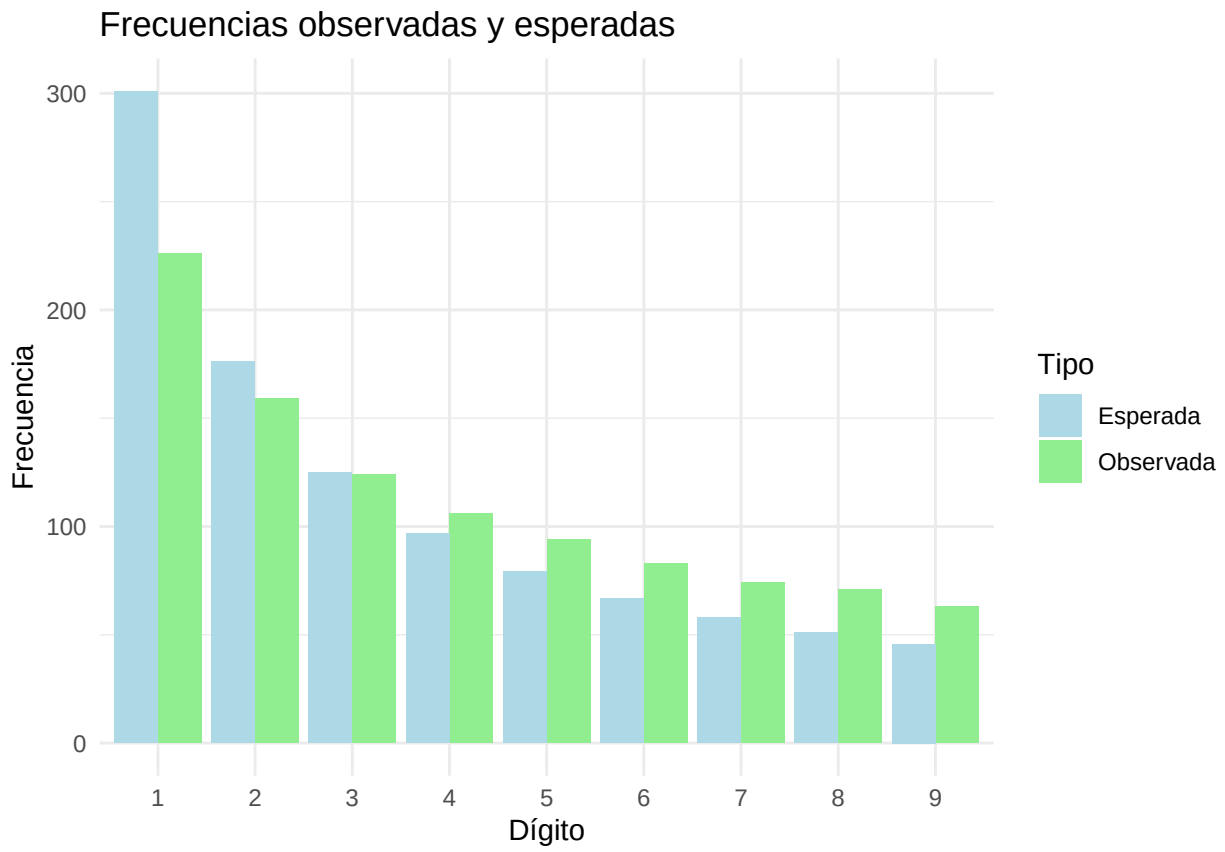
```
## [1] "p-valor:  0.664168365400183"
```

Apartado 4

Dibujamos los diagramas de frecuencias esperadas y observadas para el apartado 1:

```
frecuencias <- data.frame(
  Dígito = as.character(rep(1:9, 2)),
  Frecuencia = c(as.numeric(frecuencias_observadas), frecuencias_esperadas),
  Tipo = rep(c("Observada", "Esperada"), each = 9)
)

ggplot(frecuencias, aes(x = Dígito, y = Frecuencia, fill = Tipo)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Frecuencias observadas y esperadas", x = "Dígito", y = "Frecuencia") +
  scale_fill_manual(values = c("lightblue", "lightgreen")) +
  theme_minimal()
```



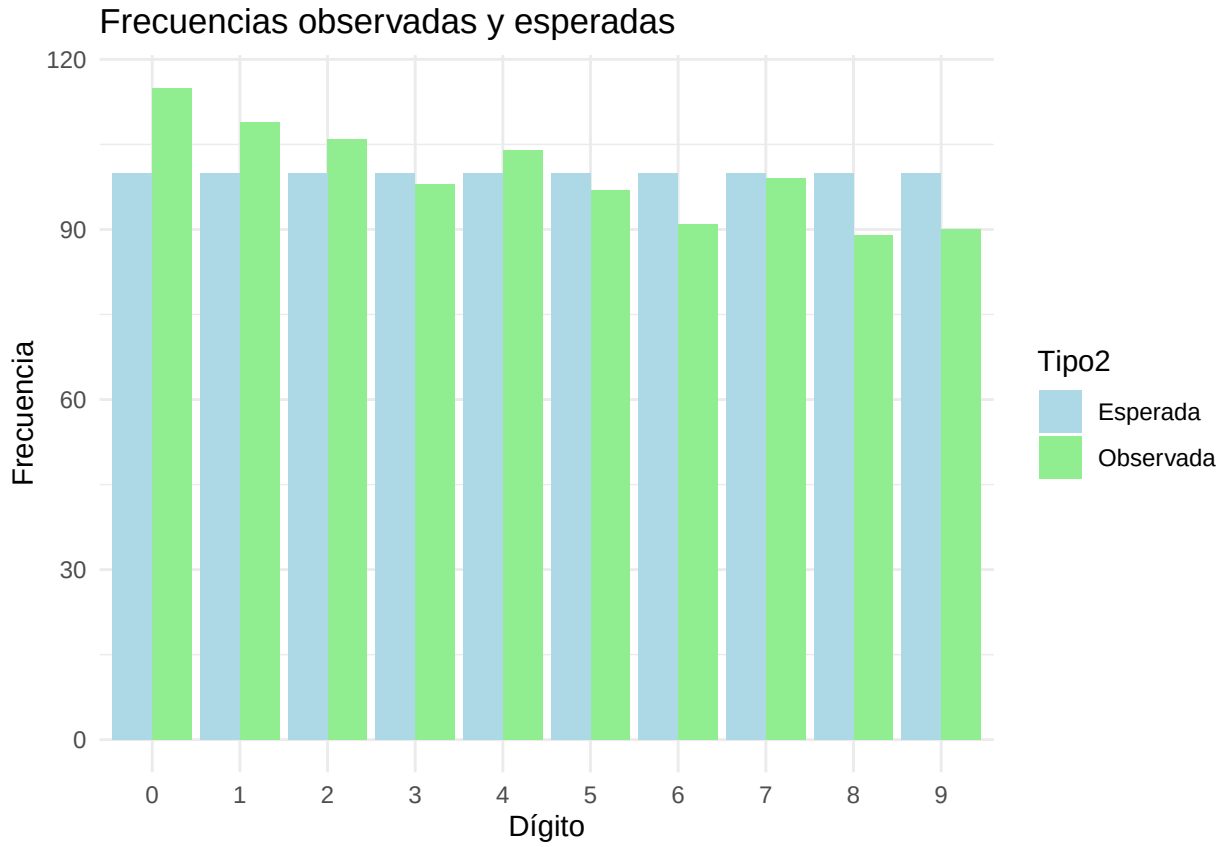
Observando la gráfica podemos concluir que los datos observados no se corresponden con los esperados, ya que hay bastante diferencia entre ellos, sobre todo entrelos digitos 1 y 9.

Ahora dibujamos los diagramas de frecuencias esperadas y observadas para el apartado 2:

```
frecuencias2 <- data.frame(
  Dígito2 = as.character(rep(0:9,2)),
  Frecuencia2 = c(as.numeric(frecuencias_observadas2), frecuencias_esperadas2),
  Tipo2 = rep(c("Observada", "Esperada"), each = 10)
```

```
)

ggplot(frecuencias2, aes(x = Dígito2, y = Frecuencia2, fill = Tipo2)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Frecuencias observadas y esperadas", x = "Dígito", y = "Frecuencia") +
  scale_fill_manual(values = c("lightblue", "lightgreen")) +
  theme_minimal()
```



Observando la gráfica podemos concluir que los datos observados no siguen una distribución uniforme discreta, ya que hay una diferencia notable entre los datos observados y los esperados.

1.3 Problema 3: Homegeneidad e independencia. (3 puntos).

Queremos analiza los resultados de aprendizaje con tres tecnologías. Para ello se seleccionan grupos de 4 Grados (Grado1, Grado2, Grado3, y Grado4) de 50 estudiantes y se les somete a evaluación después de un curso que se encuentran en los datos adjuntos `datasets/tecnologias_4_grados.csv`.

Se pide

1. Discutid si hacemos un contraste de independencia o de homogeneidad de las distribuciones de las notas por tecnología. Escribid las hipótesis del contraste.
2. Interpretad la función `chisq.test` y resolved el contraste.
3. Calculad las frecuencias teóricas como producto de los vectores marginales y calculad el estadístico de contraste y el p-valor.

1.4 Problema 4: Contraste de proporciones de dos muestras independientes. (3. puntos)

Queremos comparar las proporciones de aciertos de dos redes neuronales que detectan si una foto con un móvil de una avispa es una avispa velutina o asiática o si es una avispa común. Esta avispa en una especie invasora y peligrosa por el veneno de su picadura. Para ello disponemos de una muestra de 1000 imágenes de insectos etiquetadas como avispa velutina y no velutina.

Aquí tenéis el acceso a los datos. Cada uno está en fichero selecciona 500 fotos de de forma independiente para el algoritmo 1 y el 2. Los aciertos están codificados con 1 y los fallos con 0.

Se pide:

1. Cargad los datos desde el servidos y calcular el tamaño de las muestras y la proporción de aciertos de cada muestra.
2. Contrastad si hay evidencia de que las las proporciones de aciertos del algoritmo 1 son mayores que las del algoritmo 2. Definid bien las hipótesis y las condiciones del contraste. Tenéis que hacer el contraste con funciones de `R` y resolver el contrate con el p-valor.
3. Calculad el intervalo de confianza para la diferencia de proporciones **pág 187 tema 4: CH** que vimos de forma manual en teoría.

1.5 Problema 5 : Contraste de proporciones de dos muestras emparejadas. (2. puntos)

En el problema anterior hemos decidido quedarnos con el mejor de los algoritmos y mejorarlo. Pasamos las mismas 1000 imágenes a la version_beta del algoritmo y a la version_alpha. Aquí tenéis el acceso a los datos en el mismo orden para las 1000 imágenes. Cada uno está en fichero los aciertos están codificados con 1 y los fallos con 0.

1. Cargad los datos desde el servidos y calcular el tamaño de las muestras y la proporción de aciertos de cada muestra.
2. Contrastad si hay evidencia de que las las proporciones de aciertos del algoritmo alfa son iguales que las del algoritmo beta. Definid bien las hipótesis y las condiciones del contraste. De forma manual como se explicó en **teoría pág 246 tema 4: CH** y resolver con el p-valor.