Análisis del Play-Delay en Netflix

Luz Alba Posse & Martina Monastra

2024-10-19

Índice

1 Datos Históricos

1.1 Histograma

Para evaluar los datos históricos, calculamos la media y la varianza del play-delay y graficamos un histograma para verificar la forma de la distribución.

```r
historical_data <- read.csv("datos_historicos.csv")

mean_historical <- mean(historical_data$play.delay)
variance_historical <- var(historical_data$play.delay)

mean_historical
```

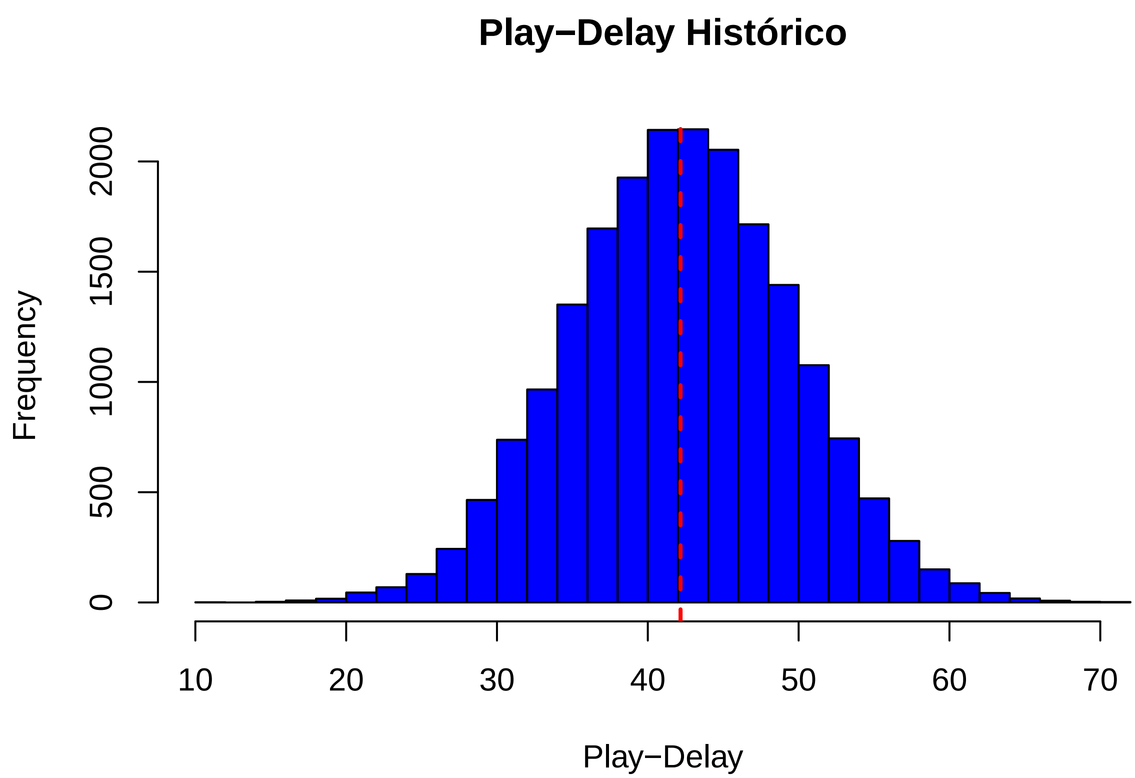
```
## [1] 42.17103
```

```r
variance_historical
```

```
## [1] 54.30905
```

```r
hist(historical_data$play.delay,
     breaks = 30,
     main = "Play-Delay Histórico",
     xlab = "Play-Delay",
     col = "blue",
     border = "black")

abline(v = mean_historical, col = "red", lwd = 2, lty = 2)
```



Como bien vemos en el gráfico, los datos parecen distribuirse de forma normal, con una media de 42.17103 y una varianza de 54.30905.

2 Grupo de Prueba

2.1 Estimación de la Media (µ) del Play-Delay para la Nueva Versión

Se estima la esperanza del "play-delay" para los nuevos 200 usuarios (grupo de prueba). La media se utiliza como estimador de µ, representando el "play-delay" medio en la nueva versión de la plataforma.

```r
new_data <- read.csv("datos_nuevos.csv")

mean_new <- mean(new_data$play.delay)

mean_new
```

```
## [1] 42.9895
```

3 Construir el test

3.1 Formulación de la Hipotesis

- Hipótesis Nula (H_0): El play-delay promedio con la nueva versión es igual al de la versión anterior. $\mu = \mu_0$, donde μ_0 es la media del play-delay histórico.
- Hipótesis Alternativa (H_1): El play-delay promedio con la nueva versión es mayor al de la versión anterior. Esto implica que $\mu > \mu_0$

Entonces, vamos a rechazar H_0 solo si la nueva versión empeora (aumenta) el play-delay.

El estadístico que utilizaremos es el Z para muestras grandes, ya que asumimos que el play-delay tiene una distribución normal y conocemos la vairanza de los datos históricos. Entonces:

$Z = \frac{\bar{X}-\mu_0}{\frac{\sigma_0}{\sqrt{n}}}$

3.2 Región de Rechazo para $\alpha = 0.05$

El nivel de significancia α es 0.05. El valor crítico $Z_{0.05}$ se obtiene de la tabla de distribución normal, que es **1.645**.

La **región de rechazo** es entonces:

$$Z > 1.645$$

Si el valor calculado de Z es mayor que 1.645, rechazamos H_0 y concluimos que la actualización aumenta el play-delay.

4 Usar el test

4.1 Decisión basada en las 200 nuevas observaciones

Calculamos el estadístico Z utilizando los datos de los 200 usuarios de prueba y lo comparamos con el valor crítico para un nivel de significancia $\alpha = 0.05$.

```r
# Parámetros
mu_0 <- mean_historical
sigma_0 <- sqrt(variance_historical)
n <- 200

# Estadístico Z
Z <- (mean_new - mu_0) / (sigma_0 / sqrt(n))

Z
```

```
## [1] 1.570643
```

El valor del estadístico Z obtenido es **1.5706434**.

El valor crítico para $\alpha = 0.05$ es **1.645**. Si el valor de Z es mayor que este valor crítico, rechazamos la hipótesis nula H_0, concluyendo que la actualización incrementa el "play-delay". Si Z es menor o igual a 1.645, no rechazamos H_0.

```r
# Valor crítico para alpha = 0.05
Z_critico <- qnorm(0.95)

# Comparar Z con el valor crítico
if (Z > Z_critico) {
  decision <- "Rechazamos H0: La actualización aumenta el play-delay."
```

```r
} else {
  decision <- "No rechazamos H0: No hay evidencia suficiente."
}

decision
```

```
## [1] "No rechazamos H0: No hay evidencia suficiente."
```

Dado que el valor del estadístico $Z = 1.5706434$ es menor que el valor crítico $Z_{0.05} = 1.645$, **no rechazamos la hipótesis nula** H_0. Por lo tanto, no tenemos evidencia suficiente para concluir que la nueva versión aumente el play-delay, y **no es necesario enviar el código a revisión**.

4.2 Regiones de rechazo para otros valores de α

El valor de α determina qué tan estrictos somos al rechazar H_0. Entonces, a medida que aumentamos α, la región de rechazo se vuelve más amplia, lo que implica que será más fácil rechazar H_0. A continuación, analizamos la región de rechazo para otros valores de α, como 0.01 y 0.1, y comparamos las decisiones.

```r
# Calcular valores críticos para diferentes alphas
Z_critico_01 <- qnorm(0.99)  # para alpha = 0.01
Z_critico_10 <- qnorm(0.90)  # para alpha = 0.1

# Comparar Z con los valores críticos
if (Z > Z_critico_01) {
  decision_01 <- "Rechazamos H0 para alpha = 0.01"
} else {
  decision_01 <- "No rechazamos H0 para alpha = 0.01"
}

if (Z > Z_critico_10) {
  decision_10 <- "Rechazamos H0 para alpha = 0.1"
} else {
  decision_10 <- "No rechazamos H0 para alpha = 0.1"
}

decision_01
```

```
## [1] "No rechazamos H0 para alpha = 0.01"
```

```r
decision_10
```

```
## [1] "Rechazamos H0 para alpha = 0.1"
```

Para un nivel de significancia $\alpha = 0.01$, el valor crítico es más alto ($Z = 2.33$), lo que hace más difícil rechazar H_0. En este caso, la decisión es **No rechazamos H0 para alpha = 0.01**.

Para $\alpha = 0.1$, el valor crítico es más bajo ($Z = 1.28$), lo que facilita rechazar H_0. En este caso, la decisión es **Rechazamos H0 para alpha = 0.1**.

Entonces, que concluimos con esto?

- **Para** $\alpha = 0.05$: No rechazamos H_0 $\rightarrow$ no hay evidencia suficiente para afirmar que la actualización aumenta el play-delay, por lo que no es necesario enviar el código a revisión.
- **Para** $\alpha = 0.01$: Somos más estrictos, y la decisión es no rechazar H_0.
- **Para** $\alpha = 0.1$: Somos menos estrictos y estamos más dispuestos a rechazar H_0.

4.3 Evaluar la decisión para diferentes valores de α

Vamos a evaluar la decisión del test para $\alpha = 0.01, 0.02, 0.03, \ldots, 0.10$. Básicamente, calculamos el valor crítico para cada α y vemos si rechazamos H_0 o no.

```r
# Definir los diferentes valores de alpha
alphas <- seq(0.01, 0.10, by = 0.01)

# Crear un dataframe para almacenar los resultados
resultados_alpha <- data.frame(alpha = alphas, Z_critico = NA, decision = NA)

# Evaluar el test para cada alpha
for (i in 1:length(alphas)) {
  Z_critico_alpha <- qnorm(1 - alphas[i])
  decision_alpha <- ifelse(Z > Z_critico_alpha, "Rechazamos H0", "No rechazamos H0")

  # Almacenar los resultados
  resultados_alpha$Z_critico[i] <- Z_critico_alpha
  resultados_alpha$decision[i] <- decision_alpha
}

# Mostrar los resultados
resultados_alpha
```

```
##    alpha Z_critico         decision
## 1   0.01  2.326348 No rechazamos H0
## 2   0.02  2.053749 No rechazamos H0
## 3   0.03  1.880794 No rechazamos H0
## 4   0.04  1.750686 No rechazamos H0
## 5   0.05  1.644854 No rechazamos H0
## 6   0.06  1.554774    Rechazamos H0
## 7   0.07  1.475791    Rechazamos H0
## 8   0.08  1.405072    Rechazamos H0
## 9   0.09  1.340755    Rechazamos H0
## 10  0.10  1.281552    Rechazamos H0
```

4.4 Encontrar el valor mínimo de α para rechazar H_0

Para calcular el p-valor, usamos la función acumulativa de la distribución normal estándar $P(Z \geq z)$.

```r
p_valor <- 1 - pnorm(Z)

p_valor
```

```
## [1] 0.05813275
```

Entonces, interpretamos que:

- Si el p-valor es muy pequeño, entonces es poco probable observar un valor como el de Z si H_0 fuera verdadera. Esto sugiere que deberíamos rechazar H_0.
- En el contexto del problema, el p-valor nos indica cuán fuerte es la evidencia de que la nueva actualización aumenta el play-delay. Un p-valor bajo significaría que es muy probable que la actualización esté afectando negativamente la experiencia del usuario.

5 Simulaciones del Error Tipo 1

5.1 Simulación de una muestra bajo H_0

Asumimos que H_0 es verdadera y simulamos una nueva muestra de 200 observaciones, aplicamos el test y tomamos una decisión.

```r
# Parámetros
n <- 200
mu_0 <- mean_historical
sigma_0 <- sqrt(variance_historical)

# Simular una muestra
set.seed(123)
muestra_simulada <- rnorm(n, mean = mu_0, sd = sigma_0)

# Calcular el estadístico Z
media_simulada <- mean(muestra_simulada)
Z_simulado <- (media_simulada - mu_0) / (sigma_0 / sqrt(n))

# Valor crítico y decisi n
Z_critico <- qnorm(0.95)
if (Z_simulado > Z_critico) {
  decision_simulada <- "Rechazamos H0"
} else {
  decision_simulada <- "No rechazamos H0"
}

# Mostrar el estadístico Z y la decisión
Z_simulado
```

```
## [1] -0.1212044
```

```r
decision_simulada
```

```
## [1] "No rechazamos H0"
```

5.2 Simulación de 10,000 muestras

Repetimos el procedimiento anterior 10,000 veces para calcular qué porcentaje de estas rechazamos H_0, obteniendo una estimación del error de tipo 1.

```r
# Parámetros
R <- 10000
rechazos_H0 <- 0

# Simulación de R muestras
set.seed(123)
for (i in 1:R) {
  muestra <- rnorm(n, mean = mu_0, sd = sigma_0)
  media_muestra <- mean(muestra)
  Z_muestra <- (media_muestra - mu_0) / (sigma_0 / sqrt(n))

  if (Z_muestra > Z_critico) {
    rechazos_H0 <- rechazos_H0 + 1
  }
}
```

```r
# Calcular el porcentaje de rechazos
porcentaje_rechazos <- (rechazos_H0 / R) * 100
porcentaje_rechazos
```

```
## [1] 4.95
```

El porcentaje de veces que rechazamos H_0 es **4.95%**.

5.3 Interpretación Teórica de α

El nivel de significancia α representa la probabilidad de cometer un error de tipo 1, es decir, rechazar H_0 cuando en realidad es verdadera. En nuestro caso, con $\alpha = 0.05$, esperamos que aproximadamente el 5% de las veces cometamos este error al realizar muchas pruebas. Lo que es consistente con el resultado obtenido en la simulación anterior.