

Permutation test using Monte Carlo approach

Medeline Amador y Luis Rojo-González

2020-02-05

Situación (este caso de estudio es de McClave et al. (1997)). Las pruebas del producto pueden automatizarse por completo o pueden realizarse con inspectores humanos o inspectores humanos con la ayuda de dispositivos mecánicos. Aunque la inspección humana es con frecuencia la alternativa más económica, puede conducir a graves problemas de error de inspección.

Para evaluar el desempeño de los inspectores en una nueva empresa, un gerente de calidad hizo que una muestra de 12 inspectores novatos evaluaran 200 productos terminados. Los mismos 200 artículos fueron evaluados por 12 inspectores experimentados.

El gerente cree que la variabilidad de los errores de inspección fue menor para los inspectores experimentados que para los inspectores novatos.

Los siguientes datos muestran el número de errores de inspección realizados por cada inspector.

```
#Datos
a = c(30,35,26,40,36,20,45,31,33,29,21,48) #novatos
b = c(31,15,25,19,28,17,19,18,24,10,20,21) #senior
datos = data.frame(experiencia = rep(c("Novato","Senior"), each = 12), fallo=c(a,b))
```

Objetivo. Comprobar si hay variabilidad en el proceso de inspección de los inspectores novatos contra los inspectores que sí tienen experiencia (senior). Para el análisis se consideran las siguientes hipótesis:

$$H_0 : \frac{\sigma_{Senior}^2}{\sigma_{Novatos}^2} = 1, H_1 : \frac{\sigma_{Senior}^2}{\sigma_{Novatos}^2} < 1$$

donde la hipótesis nula es la igualdad de las varianzas entre ambos tipos de inspectores, y la hipótesis alternativa marca la relación en donde la variabilidad de los inspectores senior es menor a la de los inspectores novatos.

Análisis exploratorio. Considerando que el objetivo es evaluar si existe variabilidad significativa entre ambos inspectores, el primer paso consta de observar los estadísticos media, mediana y la misma varianza, correspondientes a cada muestra.

```
## EDA
datos %>% group_by(experiencia) %>%
  summarise(media = mean(fallo), mediana = median(fallo),
            var = var(fallo))
```

```
## # A tibble: 2 x 4
##   experiencia media mediana  var
##   <fct>      <dbl>   <dbl> <dbl>
## 1 Novato    32.8     32    74.7
## 2 Senior    20.6    19.5  33.0
```

De esta forma es importante notar que en términos de media y mediana, ambas distribuciones parecen ser simétricas; mientras que la varianza de los inspectores novatos representa la mitad de la obtenida para los inspectores senior.

```
## Boxplot
ggplot(data = datos, aes(x = experiencia, y = fallo, colour = experiencia)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.2) +
  scale_color_manual(values = c("gray50", "orangered2")) +
```

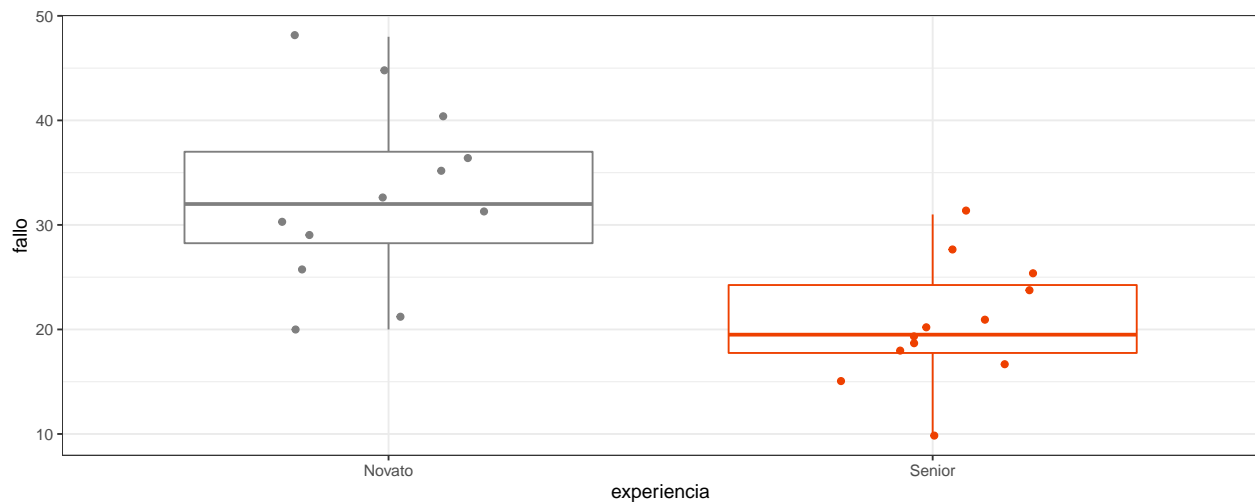


Figure 1: Gráfico de caja según experiencia del inspector.

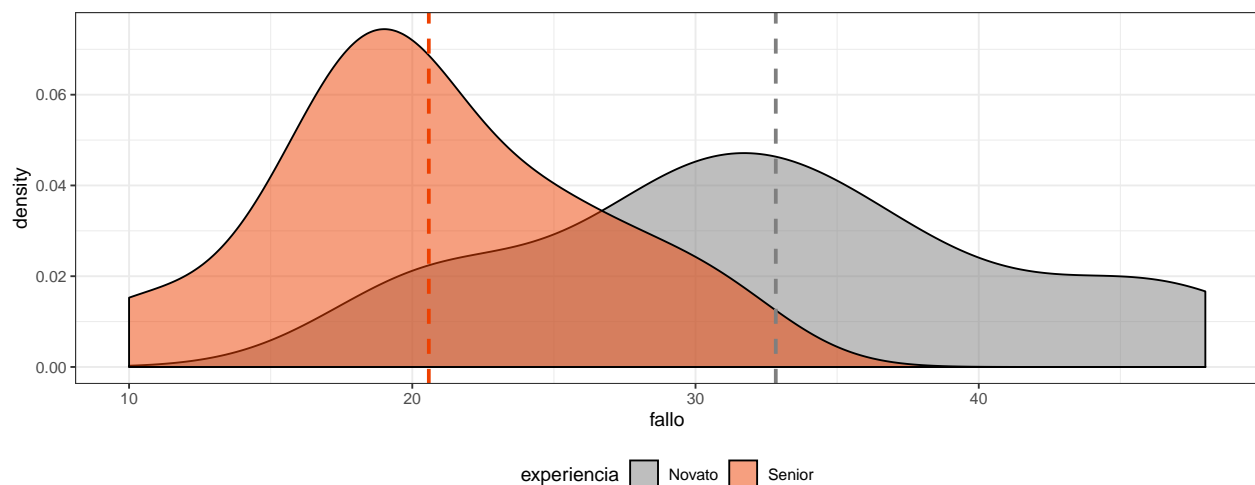


Figure 2: Gráfico de caja según experiencia del inspector.

```
theme_bw() +
theme(legend.position = "none")
```

Tal como Figura 1 muestra es importante notar que, si bien la varianza obtenida difiera para cada muestra, gráficamente esta variabilidad no es tan notoria, pues ambos gráficos se asemejarían mucho trasladando los puntos.

```
## Density plot
ggplot(data = datos, aes(x = fallo, fill = experiencia)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("gray50", "orangered2")) +
  geom_vline(xintercept = mean(datos%>%filter(experiencia=="Novato")%>%pull(fallo)),
             color = "gray50", linetype = "dashed", size = 1) +
  geom_vline(xintercept = mean(datos%>%filter(experiencia=="Senior")%>%pull(fallo)),
             color = "orangered2", linetype = "dashed", size = 1) +
  theme_bw() +
  theme(legend.position = "bottom")
```

En Figura 2 se puede apreciar que ambas distribuciones son simétricas, la distribución de fallos de los inspectores senior se encuentra más cercana a su media (línea vertical), mientras que la conclusión contraria puede obtenerse observando la distribución de los inspectores novatos.

Test de Permutaciones.

Para llevar a cabo el test de hipótesis propuesto, se realiza un test de permutaciones en donde, en primer lugar, se debe reconocer qué tipo de situación se está trabajando, es decir, si pertenece la misma cantidad de muestras a cada tipo de experiencia analizada. Por otro lado, se debe identificar la cantidad de permutaciones que se deberían hacer en caso de utilizar un test exacto, donde un número excesivo de estas permutaciones, e.g. mayor a 1,000,000, llevarían a un problema en el consumo de recursos computacionales sin ser eficiente.

```
## Tamaños muestrales
n <- tapply(datos[, "fallo"], datos[, "experiencia"], length)
n

## Novato Senior
##      12      12

N <- sum(n)
n1 <- n[1] #novato
n2 <- n[2] #senior

#Número total de Combinaciones
choose(N, n1)

## [1] 2704156
```

En particular, el caso trabajado se trata de un caso balanceado, i.e. existe la misma cantidad de muestras tanto para los inspectores novatos como para los inspectores senior, siendo 12 muestras para cada uno de ellos. Por otro lado, el número de permutaciones que generaría esta cantidad de datos corresponde a 2,704,156 permutaciones lo cual es ineficiente desde el punto de vista computacional. De acuerdo a lo anterior es idóneo trabajar el test de permutaciones con una aproximación Monte Carlo.

Aproximación Monte Carlo.

En primer lugar, un estadístico que refleje lo que se busca comprobar se debe imponer. Para este objetivo el estadístico T corresponde al mostrado en las hipótesis formuladas anteriormente.

$$T = \frac{\sigma_{Senior}^2}{\sigma_{Novatos}^2}$$

```
#Define estadístico
diff.var <- function(indexs, data) {
  var(datos[-indexs, 2]) / var(datos[indexs, 2]) # senior/novato
}

#Valor estadístico muestral observado
(dif.var.real <- diff.var(1:n1, datos))

## [1] 0.4416836
```

El estadístico muestral corresponde a 0.4417, el cual será utilizado como umbral para la comparación de las distintas permutaciones a realizar.

La aproximación Monte Carlo se realiza con 9999 permutaciones en las cuales se irá analizando el p-valor obtenido de forma gráfica y, por otro lado, se aplica la corrección al cálculo del p-valor para evitar un p-valor de 0.

Paso 1. El número de permutaciones aleatorias a generar corresponde a 9999 permutaciones.

```
# Test de Permutaciones aproximación Monte Carlo
nperm = 9999
```

Paso 2. Se genera 'dvarPerm', lo cual representa los valores de las permutaciones aleatorias obtenidas, en donde 'diff.var' corresponde a la función señalada anteriormente como el ratio entre ambas varianzas.

```
set.seed(1324)
dvarPerm = replicate(nperm, diff.var(sample(1:N, size = n1), datos))
```

Paso 3. Un resumen descriptivo de 'dvarPerm' es realizado y un el cálculo de la aproximación del p-valor es obtenido a lo largo de las permutaciones realizadas, en donde el p-valor en la aproximación Montecarlo corresponde a 0.0954.

```
# dvarPerm[1:10]
summary(dvarPerm)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.1404  0.6594  1.0043  1.2059  1.5205  6.4538

sd(dvarPerm)

## [1] 0.7902321

var(dvarPerm)

## [1] 0.6244667

p.value.mc = vector()
for (i in 1:nperm) {
  p.value.mc[i] = (sum(dvarPerm[1:i] <= dif.var.real)+1)/(i+1)
}
(p.value.mc[nperm])

## [1] 0.0923
```

Paso 4. La evolución del p-valor a medida que aumenta el número de permutaciones es realizada para validar que este se estabilice a medida que aumenta la cantidad de permutaciones realizadas. De esta forma, tal y como muestra 4, es posible apreciar que el p-valor calculado a través de esta forma se estabiliza luego de considerar 2500 permutaciones (aproximadamente), lo cual arroja un resultado final para el p-valor de 0.0923.

```
ggplot(data = data.frame(perm = c(1:nperm), p.value.mc = p.value.mc),
       aes(x = perm, y = p.value.mc)) + geom_line() +
  labs(title = "Evolución p-value en aproximación Monte Carlo",
       x = "Permutaciones", y = "p-value") +
  theme_bw()
```

Paso 5. Se grafica el estadístico del test de permutaciones para observar su comportamiento.

```
ggplot(data = data.frame(permutaciones = dvarPerm),
       aes(x = permutaciones)) +
  geom_histogram(aes(y = ..density..), alpha = 0.4, colour = "white") +
  geom_line(aes(y = ..density..), stat = 'density', colour = "green") +
  # geom_vline(xintercept = quantile(dvarPerm, probs = (1-p.value.mc[nperm]))) +
  # geom_vline(xintercept = mean(dvarPerm), colour = "red") +
  geom_vline(xintercept = dif.var.real, colour = "blue") +
  labs(title = "Distribución estadístico test de permutaciones",
       x = "Diferencia de varianzas") +
  theme_bw()
```

Conclusiones. En base al test de permutaciones mediante el uso de aproximación Monte Carlo, se ha obtenido un p-valor igual a 0.0954, por lo que con un error tipo I establecido igual a 0.1 se puede concluir que existen diferencias significativas para las varianzas de ambos grupos de inspectores. Por otro lado, los resultados para el p-valor soportan el hecho de que la utilización de una aproximación Monte Carlo con un

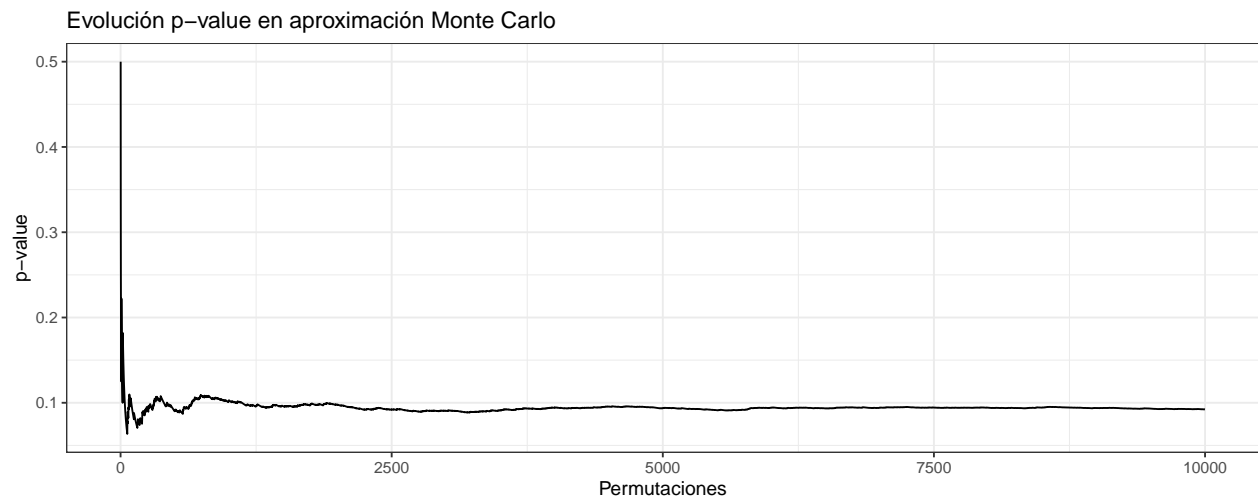


Figure 3: Evolución p-valor mediante aproximación Monte Carlo.

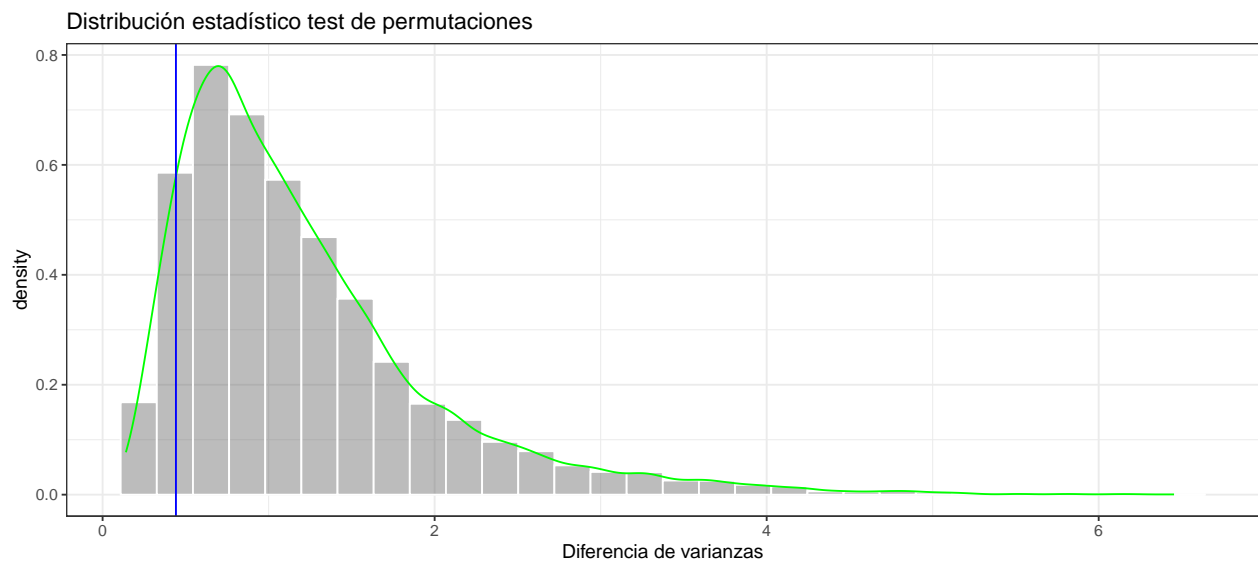


Figure 4: Evolución p-valor mediante aproximación Monte Carlo.

elevado número de permutaciones permite ser eficiente en el cálculo sin perder información para concluir. Por otro lado, en caso de haber utilizado una significancia igual a 0.05 la conclusión sería que no existen diferencias significativas en la variabilidad de ambos grupos, lo cual deja en evidencia el hecho de que se debe ser precavido e instaurar un nivel de significancia de acuerdo al nivel de riesgo que conlleva rechazar la hipótesis nula propuesta. No obstante, para este caso un nivel de significancia *alto* puede ser aplicado.