

# Chapter3

LuisF

27/5/2020

## Ejemplo webstickiness

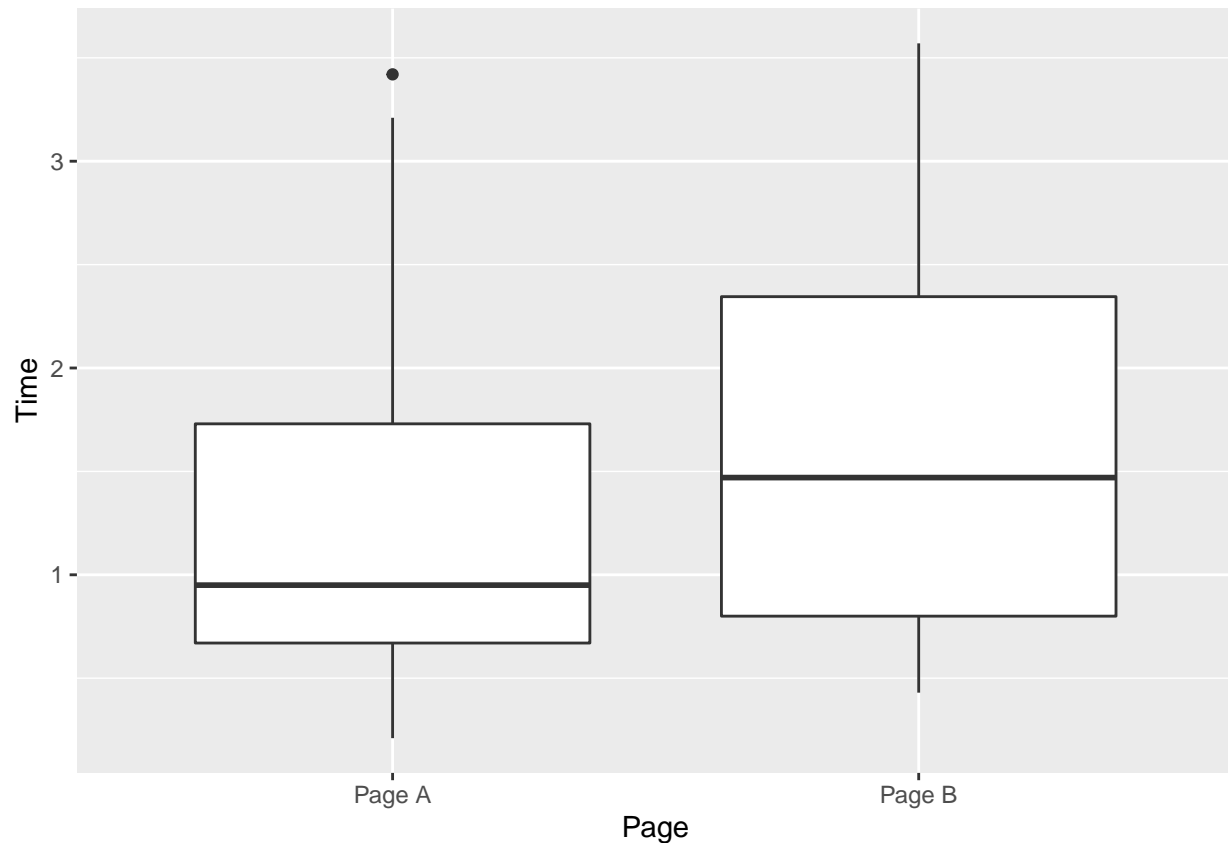
Una empresa desea saber qué diseño web le va traer más ventas. Para poder realizar esto en un tiempo corto, se va utilizar una variable proxy, ya que el hecho de intentar con un diseño y luego con otro, va tomar mucho tiempo. Una variable proxy es aquella que te va ayudar a inferir tu variable real de interes.

En este ejemplo, la variable proxy que utilizan es el tiempo que las personas pasan en la página A y B.

```
#Leyendo df  
df <- read.table('D:/Documentos/Python for Data Projects/datos_practical_statistics/web_page_data.csv',  
head(df)
```

```
##      Page Time  
## 1 Page A 0.21  
## 2 Page B 2.53  
## 3 Page A 0.35  
## 4 Page B 0.71  
## 5 Page A 0.67  
## 6 Page B 0.85
```

```
##Hay un total de 36 sesiones contadas: 21 para A y 15 para B.  
#Usando unos boxplot podemos visualizar los tiempos en cada página  
ggplot(df, aes(x=Page, y=Time)) +  
  geom_boxplot()
```



```
#restando las medias:
mean_a <- mean(df[df['Page'] == 'Page A', 'Time'])
mean_b <- mean(df[df['Page'] == 'Page B', 'Time'])
(mean_b - mean_a)*60
```

```
## [1] 21.4
```

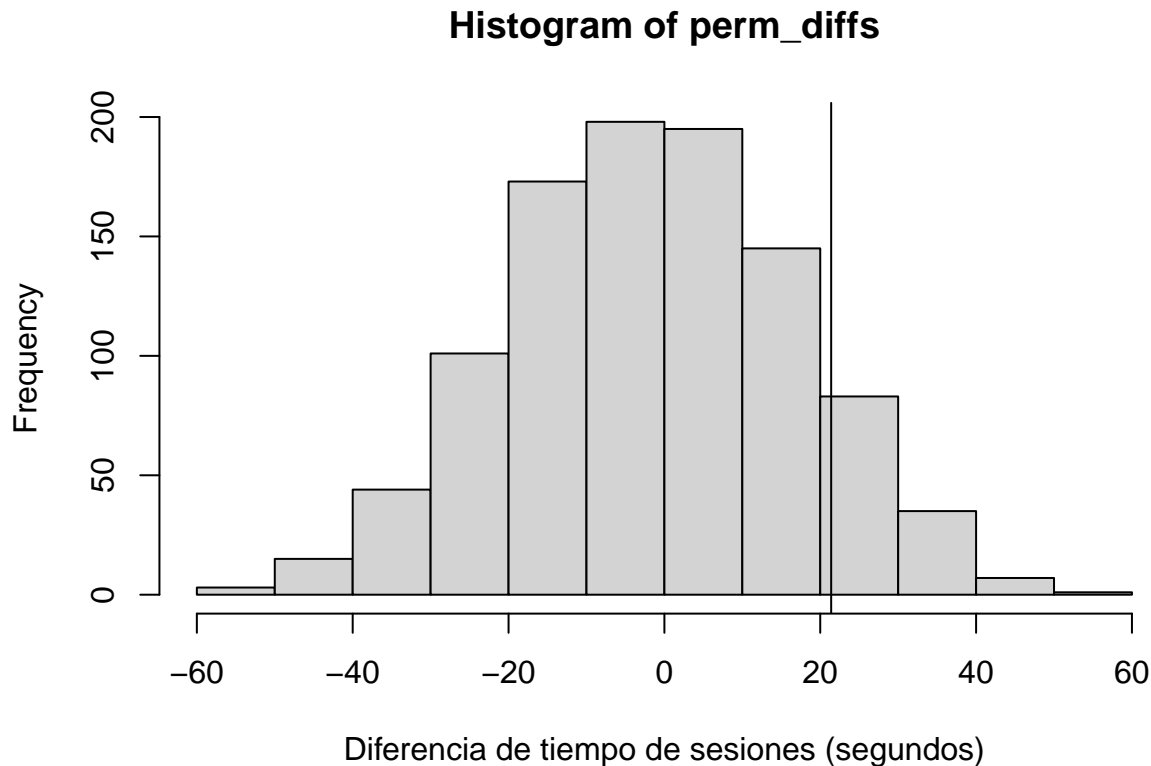
```
#La diferencia de tiempo de sesión entre B respecto a A es 21.4 segundos.
#Podrías concluir con esto que B es significativamente mejor que A? Acaso la diferencia esta dentro de
#es estadísticamente significativo?
#Vamos a aplicar un test de permutación
```

```
#Primero creamos una función para generar una permutación
perm_fun <- function(x,n1,n2) {
  n <- n1+n2
  idx_b <- sample(1:n, n1)
  #print(idx_b)
  idx_a <- setdiff(1:n, idx_b)
  #print(idx_a)
  mean_diff <- mean(x[idx_b]) - mean(x[idx_a])
  return(mean_diff)
}
```

```
#perm_fun(df[, 'Time'],21,15)
```

```
#Ahora generamos n veces esta permutación (test de permutación random):
```

```
perm_diffs <- rep(0,1000)
for (i in 1:1000) {
  perm_diffs[i] = perm_fun(df[, 'Time'], 21, 15)*60
}
hist(perm_diffs, xlab='Diferencia de tiempo de sesiones (segundos)')
abline(v=(mean_b - mean_a)*60)
```



Las permutaciones demuestran que la diferencia de promedios de tiempo caen mayormente entre -40 y 40. De manera random, esto es lo normal. La diferencia de medias entre A y B es 21, un valor que caen entre el rango de -40 y 40, por lo tanto, no es estadísticamente significativo.

Cabe mencionar que hay dos tipos de permutación random:

- Test de Permutación Exhaustiva: en lugar de unir y dividir de manera random los datos, se buscan las posibles combinaciones. Esto es practico solo para muestras pequeñas.
- Test de Permutación Bootstrap: en lugar de realizar el sampleo sin reemplazo, se hace con reemplazo.

El autor menciona que la diferencia de estas dos no tiene consecuencias mayores para la ciencia de datos...

### Estadísticamente significativo y el P-Value/Statistical Significance and P-Value

Es el método utilizado por las personas que se dedican a la estadística para demostrar que la diferencia entre A y B es realmente significativa. Es decir, la diferencia que lo random puede llegar a decir entre A y B, si tendra un impacto significativo en los resultados, o no.

### Ejemplo de tasas de conversión entre páginas de e-commerce

Se tiene la siguiente información y se quiere saber

```
df <- data.frame(
  Outcome = c("Conversion", "No Conversion"),
  Price_A = c(200, 23539),
  Price_B = c(182, 22406)
)
df
```

```
##      Outcome Price_A Price_B
## 1   Conversion     200     182
## 2 No Conversion  23539  22406
```

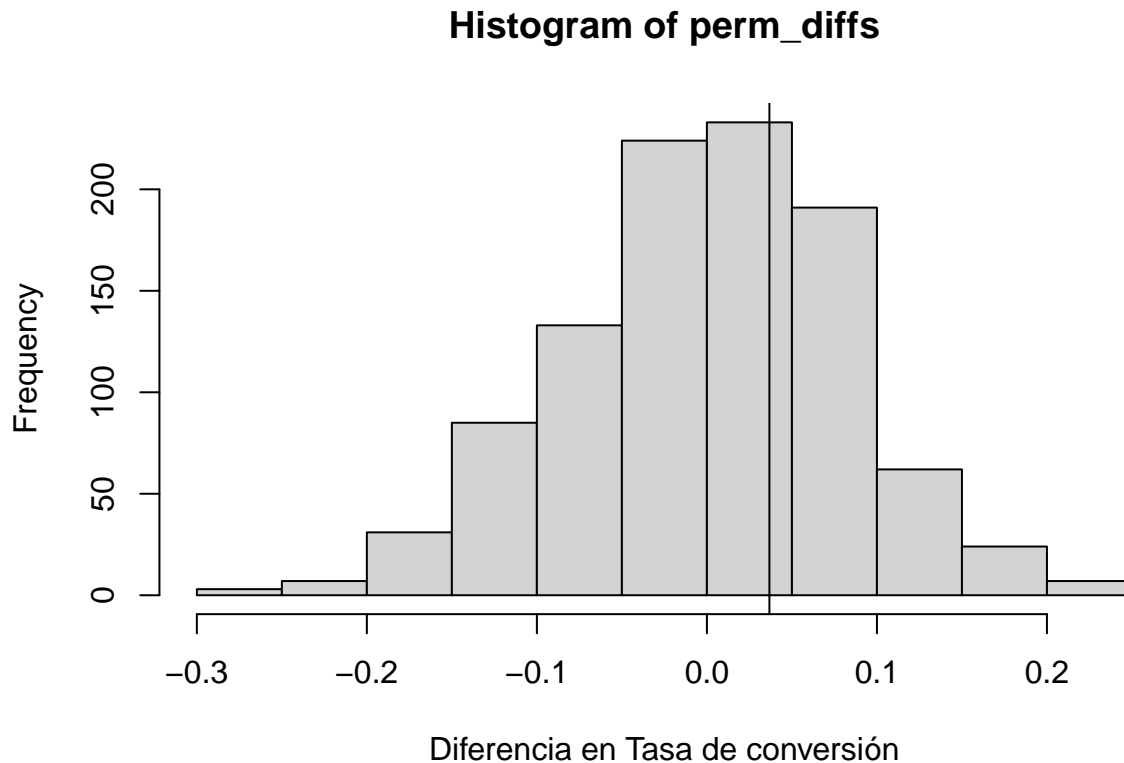
El precio A convierte casi un 5% más que el precio B:

```
tasa_conversion_a <- (df$Price_A[1]/sum(df$Price_A))*100
tasa_conversion_b <- (df$Price_B[1]/sum(df$Price_B))*100
tasa_conversion_a - tasa_conversion_b
```

```
## [1] 0.03675791
```

Sin embargo, aunque aca tenemos más de 45,000 data points y puede ser considerado bigdata (lo dudo...), aun asi necesitamos usar un test estadístico, ya que las tasas de conversión son muy bajas (menor a 1%). Hipótesis nula: la diferencia de tasas es mayor a 5% el 95% de las veces

```
obs_pct_diff = tasa_conversion_a - tasa_conversion_b
perm_diffs <- rep(0, 1000) #arreglo con puros 0 donde se guardaran los resultados
#1.- crea una urna que represente a todos los puntos de conversión y no conversión (382, 45845)
conversion <- c(rep(0, 45945), rep(1, 382))
for (i in 1:1000){
  perm_diffs[i] = 100*perm_fun(conversion, 23739, 22588)
}
hist(perm_diffs, xlab='Diferencia en Tasa de conversión')
abline(v=obs_pct_diff)
```



Al mirar este histograma es difícil saber con exactitud cuanto % es que el resultado es mayor a 0.0368. Para esto se puede utilizar el p-value:

```
mean(perm_diffs > obs_pct_diff)
```

```
## [1] 0.349
```

Esto significa que el 37.3% de las ocasiones, la tasa de conversión será mayor a 0.0368.

Ahora, se puede hacer un test binomial para obtener más fácil un p-value:

```
prop.test(x=c(200,182), n=c(23739,22588), alternative = 'greater')
```

```
##
## 2-sample test for equality of proportions with continuity correction
##
## data:  c(200, 182) out of c(23739, 22588)
## X-squared = 0.14893, df = 1, p-value = 0.3498
## alternative hypothesis: greater
## 95 percent confidence interval:
## -0.001057439 1.000000000
## sample estimates:
##      prop 1      prop 2
## 0.008424955 0.008057376
```

```
##?prop.test
```

Notas:

- p-value: es la frecuencia en la que un modelo random produce resultados más extremos de los esperados

o ya conocidos. En el ejemplo de las tasas de conversión, vemos que el p-value es = a 37% aprox. Nuestra medida observada fue que A convertía un casi 5% más que B. La hipótesis nula es que fueran iguales las tasas de conversión y la alternativa fue que alguno de esas dos páginas generara una diferencia de conversión arriba del 5%. El 37% del p-value es la frecuencia de que alguna pág sea mayor un %5 que la otra. Por lo tanto se acepta la hipótesis nula.

## t-Tests, Multiple Tests

- Todos los experimentos estadísticos necesitan una métrica para evaluarlo, y ver si su comportamiento esta bajo el rango random o es extremo.
- Bonferroni adjustment: dividir tu alpha (critical value) / n observaciones

Es demasiado complejo como estadístico, asegurar que tus experimentos son realmente fehacientes. Para lo que concierne a un científico de datos, aca tienes que echar un ojo:

- For predictive modeling, the risk of getting an illusory model whose apparent efficacy is largely a product of random chance is mitigated by crossvalidation (see “Cross-Validation”), and use of a holdout sample.
- For other procedures without a labeled holdout set to check the model, you must rely on: Awareness that the more you query and manipulate the data, the greater the role that chance might play; and Resampling and simulation heuristics to provide random chance benchmarks against which observed results can be compared.

Un poco de info general: <https://towardsdatascience.com/statistical-tests-when-to-use-which-704557554740>  
A null hypothesis, proposes that no significant difference exists in a set of given observations. For the purpose of these tests in general:

- Null: Given two sample means are equal
- Alternate: Given two sample means are not equal

## Grados de libertad

<https://blog.minitab.com/blog/statistics-and-quality-data-analysis/what-are-degrees-of-freedom-in-statistics> Aca se explica de manera buena cómo es que en un modelo de regresión múltiple, por ejmplo, si tienes más variables que observaciones, no podrás generar grados de libertad y por lo tanto, tampoco p-values ni errores...

## ANOVA: análisis de la varianza entre varios grupos

Con base en el ejemplo de tiempo dentro de 4 páginas web, en lugar de preocuparse por definir tests entre pares (pairwise) y poder caer en el bias del Multiple Test y el dicho de ‘expresime los datos y vas a obtener algo’ (Error tipo 1, falsos positivos), ANOVA te permite realizar la comparación de varianza entre los n grupos en un solo experimento.

```
df <- read.table('D:/Documentos/Python for Data Projects/datos_practical_statistics/four_sessions.csv',
head(df)
```

```
##      Page Time
## 1 Page 1   164
## 2 Page 2   178
## 3 Page 3   175
## 4 Page 4   155
## 5 Page 1   172
## 6 Page 2   191
```

```
library(lmPerm)
summary(aovp(Time ~ Page, df))
```

```
## [1] "Settings: unique SS "
```

```
## Component 1 :
##           Df R Sum Sq R Mean Sq Iter Pr(Prob)
## Page1      3   831.4   277.13 2783  0.09163 .
## Residuals  16  1618.4   101.15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Para obtener el F-statistic:
#summary(aov(Time ~ Page, df))
```

El p-value es = a 6% aprox. Por lo tanto la varianza entre los tiempos activos de estas páginas es prácticamente igual en un modelo random.

Notas para ANOVA: - Hay ciertas suposiciones o no se si llamarles condiciones: los grupos se distribuyen normalmente, hay similitud en la varianza (o promedio) de cada grupo, y los grupos son independientes. - Ho: las medias de todos los grupos son iguales. Ha: al menos una media es significativamente distinta, 'no todas las medias son distintas'. ¿Cómo sabes qué medias son las distintas? Se puede aplicar una 'prueba de Tukey' (procedimientos post hoc) - Hay ANOVA two-ways en donde puedes examinar la varianza de dos factores.

### Multi-arm bandit algorithm

Es un método que permite comparar algun parámetro de varios factores de manera más óptima y menos rígida que la estadística tradicional. Este concepto viene de una máquina traga-monedas con varias palancas por ejmplo. Supon que estas jugando en una máquina con 3 palancas (A,B y C) y quieres saber si alguna de ellas tiene tasa de retorno, entonces, tiras 50 veces cada una de ellas y la palanca 'A' te da 2 veces más victorias que B y C. Conslusiones extremas serían asumir que A tiene más tasa de retorno solo por este primer acercamiento y entonces solo jalar A. Otra sería asumir que las 3 palancas no tienen suficiente diferencia y entonces sigues jalando las 3 por igual.

Este algoritmo es 'híbrido' y te permite jalar un poco más A (aproevchando los primeros resultados) pero no se deja atrás B y C, estas, se siguen jalando pero a menor medida. Ahora, 'on the run', puedes ir evaluando si A sigue con resultados mayores o alguna de las otras palancas tuvo un repunte.

Lo interesante es saber cómo modificar estos 'rates' para elegir un factor y cuándo se deben de cambiar: - Epsilon - Thompson's sampling (bayesiano)

### Power y Sample Size: cómo saber cuantos datos necesitas para hacer tus experimentos

- Sample size: para sacar el tamaño de tu muestra es necesario tomar como base el 'effect size', el cual, es lo que esperas que B tenga más o menos que A. Por ejemplo,tienes una nueva Ad y solo las implementarás si tiene un click-rate de 10% más que tu Ad actual.
- Power: es la probabilidad de encontrar un effect size con una muestra x.