

In [ ]:

```
# Initialize Otter
import otter
grader = otter.Notebook("lab06.ipynb")
```



**UNIVERSIDAD  
DE ANTIOQUIA**  
1803

## Fundamentos en ciencias de datos

### Laboratorio 6:

¡Bienvenidos al Laboratorio 6!

Después de una introducción tan extensa a la programación para la ciencia de datos, finalmente pasamos a la sección del curso donde podemos aplicar nuestras nuevas habilidades para responder preguntas reales.

En este laboratorio, usaremos técnicas de prueba que se introdujeron en la conferencia para probar la idea del toque terapéutico, la idea de que algún practicante puede sentir y masajear su campo de energía humano.

In [25]:

```
# Ejecute esta celda, pero no la cambie.

# Estas líneas importan los módulos Numpy y Datascience.
import numpy as np
from datascience import *

# Estas líneas hacen algo de magia en la trama
import matplotlib
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
import warnings
warnings.simplefilter('ignore', FutureWarning)
from matplotlib import patches
from ipywidgets import interact, interactive, fixed
import ipywidgets as widgets
```

Ingresa tu nombre con apellido y número de carnet en las variables dadas en la celda siguiente: Ejemplo:

```
NombreApellidos="Lisa Simpson"
NumeroCarnet="27182818"
Email="lisa.simpson@udea.edu.co"
```

In [26]:

```
# BEGIN SOLUTION NO PROMPT
NombreApellidos="Lisa Simpson"
NumeroCarnet="27182818"
Email="homero.simpson@udea.edu.co"
# END SOLUTION
""" # BEGIN PROMPT
NombreApellidos="..."
NumeroCarnet=".."
Email="..."
""" # END PROMPT
```

Out [26]:

```
' # BEGIN PROMPT\nNombreApellidos="..." \nNumeroCarnet=".." \nEmail="..." \n'
```

In [ ]:

```
grader.check("qt")
```

## 1. ¿Qué es el Toque Terapéutico?

El Toque Terapéutico (TT) es la idea de que todos pueden sentir el Campo de Energía Humana (HEF) alrededor de los individuos. Quienes practican TT han descrito los HEF de diferentes personas como "cálidos como gelatina" y "táctiles como caramelo".

La TT fue una técnica popular utilizada a lo largo del siglo XX que se promocionó como una excelente manera de equilibrar la salud de una persona. Ciertos profesionales afirman que tienen la capacidad de sentir el HEF y masajearlo para promover la salud y la relajación de las personas.

### Emily Rosa

[Emily Rosa \(https://en.wikipedia.org/wiki/Emily\\_Rosa\)](https://en.wikipedia.org/wiki/Emily_Rosa) era una estudiante de 4º de primaria que estaba muy familiarizada con el mundo de la TT, gracias a sus padres, que eran a la vez médicos y escépticos de la TT.

Para su proyecto de feria de ciencias de cuarto grado, Emily decidió probar si los practicantes de TT realmente podían interactuar con el HEF de una persona. Más tarde publicó su trabajo en TT, convirtiéndose en la persona más joven en publicar un artículo de investigación en una revista médica revisada por pares.

### El experimento de Emily

El experimento de Emily fue limpio, simple y efectivo. Debido a las ocupaciones de sus padres en el campo médico, tenía amplio acceso a personas que afirmaban ser practicantes de TT.

Emily tomó 21 practicantes de TT y los usó para su experimento científico. Tomaría a un practicante de TT y le pediría que extendiera sus manos a través de una pantalla (a través de la cual no puede ver). Emily estaría del otro lado y lanzaría una moneda justa. Dependiendo de cómo cayera la moneda, extendía su mano izquierda o su mano derecha. El practicante de TT tendría entonces que responder qué mano extendió Emily. Si un practicante realmente pudiera interactuar con el HEF de una persona, se esperaría que respondiera correctamente.

En total, a través de 210 muestras, el practicante eligió la mano correcta el 44% de las veces.

El objetivo principal de Emily aquí era probar si las conjeturas de los practicantes de TT eran aleatorias, como el lanzamiento de una moneda. En la mayoría de los experimentos médicos, esta es la norma. **Queremos probar si el tratamiento tiene un efecto o no, no si el tratamiento realmente funciona o no.**

Ahora comenzaremos a formular este experimento en términos de la terminología que aprendimos en este curso.

**Pregunta 1.1:** Describa el [modelo \(https://inferentialthinking.com/chapters/11/1/Assessing\\_a\\_Model.html\)](https://inferentialthinking.com/chapters/11/1/Assessing_a_Model.html) de Emily sobre la probabilidad de que los practicantes de TT elijan la mano correcta. ¿Qué modelo alternativo pretende desacreditar su modelo?

Si puede, consulte con sus compañeros, o con los profesores de su laboratorio para llegar a una conclusión.

*Agregue una celda tipo markdown y copie su respuesta*

**Pregunta 1.2:** Recuerde que el practicante obtuvo la respuesta correcta el 44% (0,44) de las veces. Según el modelo de Emily, en promedio, ¿qué proporción de veces esperamos que el practicante adivine la mano correcta? Asegúrate de que tu respuesta sea un número entre 0 y 1.

In [28]:

```
# BEGIN SOLUTION NO PROMPT
# Calculando la proporción esperada de veces que el practicante adivina correctamente
proporcion_correcta = 0.50

# END SOLUTION
""" # BEGIN PROMPT
proporcion_correcta = ...
proporcion_correcta
""" # END PROMPT
```

Out[28]:

```
' # BEGIN PROMPT\nproporcion_correcta = ...\nproporcion_correcta\n'
```

In [ ]:

```
grader.check("q2")
```

El objetivo ahora es ver si nuestra desviación de esta proporción esperada de respuestas correctas se debe a algo más que al azar.

**Pregunta 1.3:** Generalmente utilizamos una estadística para ayudar a determinar hacia qué modelo apunta la evidencia. ¿Qué estadística podemos usar para comparar los resultados del modelo de Emily con lo observado? Asigne `valid_stat` a una matriz de números enteros que representan estadísticas de prueba que Emily puede usar :

1. La diferencia entre el porcentaje correcto esperado y el porcentaje correcto real
2. La diferencia absoluta entre el porcentaje correcto esperado y el porcentaje correcto real
3. La suma del porcentaje correcto esperado y el porcentaje correcto real

**NOTA:** ¡Asegúrese de usar `make_array` para crear su matriz de números enteros!.

La idea, es que el estudiante cree un arreglo, usando 'make\_array' seleccionando 1 o varias de las opciones, por ejemplo `make_array(3)` si considera que solamente es la tercer opción o `make_array(1,3)` si considera que son las opciones 1 y 3.

*Sugerencia: ¿Cuál debería ser el dominio (posibles valores de x) para la distribución de nuestras estadísticas de prueba?*

In [30]:

```
# BEGIN SOLUTION NO PROMPT

# Asignamos `valid_stat` a una matriz de números enteros representando las estadísticas de prueba
valid_stat = make_array(2)

# END SOLUTION
""" # BEGIN PROMPT
valid_stat = ...
valid_stat
""" # END PROMPT
```

Out[30]:

```
' # BEGIN PROMPT\nvalid_stat = ...\nvalid_stat\n'
```

In [ ]:

```
grader.check("q3")
```

**Pregunta 1.4:** ¿Por qué la estadística de la Pregunta 1.3 es la opción adecuada para comparar resultados en el experimento de Emily? ¿Cómo se relaciona con los modelos que definió en la Pregunta 1.1?

Agregue una celda tipo markdown y copie su respuesta

**Pregunta 1.5:** Defina la función "estadist" que toma una **proporción esperada** y una **proporción medida**, y devuelve el valor de la estadística elegida en la Pregunta 1.3. Suponga que el argumento toma proporciones, pero devuelva su respuesta como un porcentaje.

*Pista:* Recuerde que estamos pidiendo un **porcentaje**, no una proporción.

In [32]:

```
# BEGIN SOLUTION NO PROMPT
def estadist(proporcion_esperada, proporcion_medida):
    estadistica_valor = abs(proporcion_esperada - proporcion_medida)
    return estadistica_valor*100

# END SOLUTION
""" # BEGIN PROMPT
def estadist(proporcion_esperada, proporcion_medida):
    ...
""" # END PROMPT
```

Out[32]:

```
' # BEGIN PROMPT\ndef estadist(proporcion_esperada, proporcion_medida):\n    ...\n'
```

In [ ]:

```
grader.check("q5")
```

**Pregunta 1.6:** Utilice la función recién definida para calcular la estadística observada en el experimento de Emily.

In [34]:

```
# BEGIN SOLUTION NO PROMPT

# Calcular la estadística observada para el experimento de Emily
observaciones_emily = 0.44
estadistica_observada = estadist(observaciones_emily,0.5)

# END SOLUTION
""" # BEGIN PROMPT
estadistica_observada = ...
estadistica_observada
""" # END PROMPT
```

Out[34]:

```
' # BEGIN PROMPT\nestadistica_observada = ...\nestadistica_observada\n'
```

In [ ]:

```
grader.check("q6")
```

**¿Esta estadística observada es consistente con lo que esperamos ver bajo el modelo de Emily?**

Para responder a esta pregunta, debemos simular el experimento como si el modelo de Emily fuera correcto y calcular nuestra estadística para cada simulación.

## proporciones\_de\_muestra

`sample_proportions` se puede utilizar para realizar muestras aleatorias de varias categorías cuando se conoce la proporción de puntos de datos que se espera que caigan en cada categoría. `sample_proportions` toma dos argumentos: el tamaño de la muestra y una serie de proporciones correspondientes a cada categoría de la población (debe sumar 1).

Considere lanzar una moneda justa, donde los dos resultados (la moneda sale cara y la moneda sale cruz) ocurren con la misma probabilidad. Esperamos que la mitad de todos los lanzamientos de monedas arrojen cara y la otra mitad de todos los lanzamientos de monedas arrojen cruz.

Ejecute la siguiente celda para ver la simulación de 10 lanzamientos de una moneda justa. Sea el primer elemento de "coin\_proportions" la proporción de caras y el segundo elemento de "coin\_proportions" la proporción de cruces.

Observe lo que sucede cuando ejecutas esta celda varias veces: la proporción de lanzamientos de monedas que salen cara y cruz parece cambiar, ya que estás simulando lanzar 10 monedas cada vez.

In [36]:

```
coin_proportions = make_array(0.5, 0.5)
ten_flips = sample_proportions(10, coin_proportions)
ten_flips
```

Out[36]:

```
array([ 0.2,  0.8])
```

`sample_proportions` devuelve una matriz que tiene la misma longitud que la matriz de proporciones que se pasa. Contiene la proporción de cada categoría que aparece en la muestra.

En nuestro ejemplo, el primer elemento de "ten\_flips" es la proporción simulada de caras y el segundo elemento de "ten\_flips" es la proporción simulada de cruces.

In [37]:

```
simulated_proportion_heads = ten_flips.item(0)
simulated_proportion_tails = ten_flips.item(1)

print("En nuestra simulación, " + str(simulated_proportion_heads) + " de lanzamientos que fueron caras
y " \
    + str(simulated_proportion_tails) + " de lanzamientos que fueron cruz.")
```

En nuestra simulación, 0.2 de lanzamientos que fueron caras y 0.8 de lanzamientos que fueron cruz.

**Pregunta 1.7:** Para comenzar a simular, debemos comenzar creando una representación del modelo de Emily para usarla en nuestra simulación. Esta será una matriz con dos elementos. El primer elemento debe ser la proporción de veces que un practicante de TT elige la mano correcta, asumiendo que el modelo de Emily era correcto. El segundo elemento debe ser la proporción de veces, bajo el mismo supuesto, que el practicante de TT elige la mano incorrecta. Asigne `model_proportions` a esta matriz.

Después de esto, podemos simular 210 elecciones de manos, como Emily evaluó en la vida real, y encontrar una estadística única para resumir este ejemplo de la simulación. Utilice la función `sample_proportions` y asigne la **proporción de elecciones de manos correctas** (de 210) a `simulation_proportion_correct`. Por último, utilice su función `estadist` para asignar `one_statistic` al valor de la estadística para esta simulación.

*Sugerencia:* El uso de `sample_proportions` se puede encontrar en la [Referencia de Python \(https://www.data8.org/fa23/reference/\)](https://www.data8.org/fa23/reference/).

In [38]:

```
# BEGIN SOLUTION NO PROMPT

# Esto guarda el estado aleatorio de nuestro código para que podamos
# generamos los mismos números cada vez que ejecutamos el código.
# Por favor no cambie esta línea.
np.random.seed(16)
# Representación del modelo de Emily
model_proportions = make_array(0.44, 0.56)
# Simulación de 210 elecciones de manos
num_events = 210
simulation_results = sample_proportions(num_events, model_proportions)
# Asignar el valor de la estadística para esta simulación
simulation_proportion = simulation_results[0]
one_statistic = estadist(0.5, simulation_proportion)
one_statistic

# END SOLUTION

""" # BEGIN PROMPT
'''Esto guarda el estado aleatorio de nuestro código para que podamos
generamos los mismos números cada vez que ejecutamos el código.
Por favor no cambie esta línea.'''
np.random.seed(16)
num_events = ...
model_proportions = ...
simulation_results = ...
simulation_proportion = ...
one_statistic = ...
one_statistic

""" # END PROMPT
```

Out[38]:

```
" # BEGIN PROMPT\n'''Esto guarda el estado aleatorio de nuestro código para que podamos\ngeneramos los\nmismos números cada vez que ejecutamos el código.\nPor favor no cambie esta\nlínea.'''\nnp.random.seed(16)\nnum_events = ... \nmodel_proportions = ... \nsimulation_results = ... \nsimulation_proportion = ... \none_statistic = ... \none_statistic\n\n"
```

In [ ]:

```
grader.check("q7")
```

**Pregunta 1.8:** Veamos ahora cómo es realmente la distribución de las estadísticas según el modelo de Emily.

Defina la función `simulation_and_statistic` para incluir el arreglo `model_proportions` y la proporción esperada de veces que un practicante de TT adivinaría una mano correctamente según el modelo de Emily. La función debería simular que Emily realiza el experimento 210 veces y devolver la estadística de esta simulación.

*Pista:* Esto debería seguir el mismo patrón que el código que hiciste en el problema anterior.

In [40]:

```
# BEGIN SOLUTION NO PROMPT

def simulation_and_statistic(model_proportions_, experimentos=210):
    # Generar 210 resultados aleatorios basados en las proporciones del modelo
    resultados = np.random.choice([0, 1], size=experimentos, p=model_proportions_)

    proporción_exitos = np.mean(resultados)
    return proporción_exitos

# END SOLUTION

""" # BEGIN PROMPT
def simulation_and_statistic(model_proportions, expected_proportion_correct):
    '''Simulates 210 TT hand choices under Emily's model.
    Returns one statistic from the simulation.'''
    ...

""" # END PROMPT
```

Out[40]:

```
" # BEGIN PROMPT\ndef simulation_and_statistic(model_proportions, expected_proportion_correct):\n    '\n    ''Simulates 210 TT hand choices under Emily's model.\n    Returns one statistic from the\n    simulation.'''\n    ...\n    \n"
```

In [ ]:

```
grader.check("q8")
```

In [42]:

```
num_repetitions = 1000

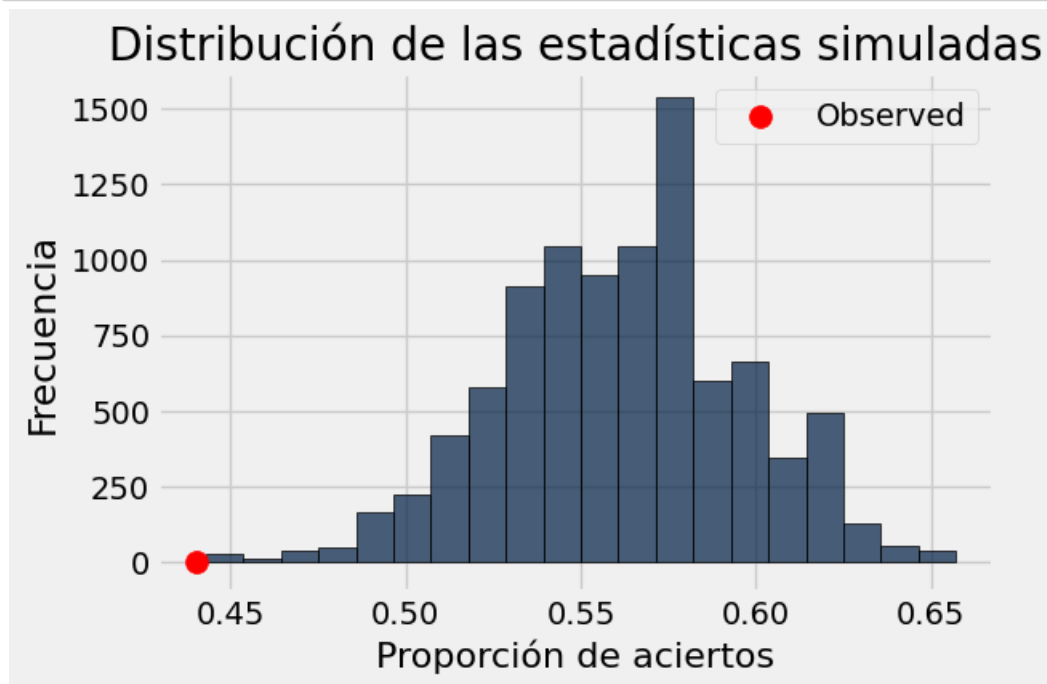
simulated_statistics = make_array()
observed_statistic = 0.44

for _ in range(num_repetitions):
    stat = simulation_and_statistic(model_proportions)
    simulated_statistics=np.append(simulated_statistics,stat)
```

Veamos la distribución de las estadísticas simuladas bajo el modelo de Emily y comparemos visualmente dónde se encuentra la estadística observada en relación con las estadísticas simuladas.

In [43]:

```
# Crear una tabla con las estadísticas simuladas
t = Table().with_column('Simulated Statistics', simulated_statistics)
# Graficar la distribución de las estadísticas simuladas
t.hist(bins=20, edgecolor='black')
plt.title('Distribución de las estadísticas simuladas')
plt.xlabel('Proporción de aciertos')
plt.ylabel('Frecuencia')
# Marcar la estadística observada en la gráfica
plt.scatter(observed_statistic, 0, color='red', s=100, zorder=2, label='Observed')
plt.legend()
plt.ylim(-1)
plt.show()
```



Podemos evaluar visualmente si la estadística observada es consistente con el modelo de Emily. Como valores más altos de la estadística de prueba favorecen el modelo alternativo (donde la probabilidad de adivinar correctamente la mano no es exactamente del 50%), podemos formalizar nuestro análisis calculando la proporción de estadísticas simuladas que son iguales o mayores a la observada en el experimento. Esta proporción representa el área bajo la curva a la derecha de la estadística de prueba observada. Si esta área es lo suficientemente pequeña, concluiremos que los datos observados son inconsistentes con el modelo simulado.

Aquí está el [enlace \(https://inferentialthinking.com/chapters/11/1/Assessing\\_a\\_Model.html\)](https://inferentialthinking.com/chapters/11/1/Assessing_a_Model.html) a la sección del libro de texto.

**Pregunta 1.9:** Calcule la proporción de estadísticas simuladas en la Pregunta 1.8 mayor o igual a la estadística observada.

*Pista:*

1) El uso de `np.count_nonzero` se puede encontrar [aquí \(http://data8.org/fa23/reference\)](http://data8.org/fa23/reference).

2) Considere el uso de `np.array()`



In [44]:

```
# BEGIN SOLUTION NO PROMPT

# Calcular la proporción de simulaciones mayores o iguales a la estadística observada

proportion_greater_equal = np.count_nonzero(np.array(simulated_statistics) >= observed_statistic) / len(simulated_statistics)
proportion_greater_equal

# END SOLUTION
""" # BEGIN PROMPT
proportion_greater_equal = ...
proportion_greater_equal
""" # END PROMPT
```

Out[44]:

```
' # BEGIN PROMPT\nproportion_greater_equal = ...\nproportion_greater_equal\n'
```

In [ ]:

```
grader.check("q9")
```

Por convención, a menudo comparamos la proporción que acabamos de calcular con 0,05. Si la proporción de estadísticas simuladas mayores o iguales a las estadísticas observadas es suficientemente pequeña (menor o igual a 0,05), entonces esto es evidencia en contra del modelo de Emily. Conceptualmente, se puede pensar en este caso como el caso en el que menos del 5% de los valores simulados están tan o más lejos de lo que esperábamos. Si este no es el caso, no tenemos ningún motivo para dudar del modelo de Emily.

Esto debería ayudarte a sacar tus propias conclusiones sobre el experimento de Emily Rosa.

El contacto terapéutico dejó de utilizarse después de este experimento, que finalmente fue aceptado en una de las principales revistas médicas. Los practicantes de TT respondieron y acusaron a Emily y su familia de alterar los resultados, mientras que algunos afirmaron que el mal humor espiritual de Emily hacia el contacto terapéutico hacía difícil leer su HEF. Sea lo que sea, el experimento de Emily es un ejemplo clásico de cómo cualquiera, con los recursos adecuados, puede probar lo que quiera.

**Pregunta 1.10:** Ahora, tómate un tiempo para reflexionar sobre las preguntas a continuación y luego discútelas con tus compañeros o echa un vistazo a las discusiones en la publicación educativa de este laboratorio.

1. ¿Son los datos más consistentes con el modelo de Emily (los practicantes adivinaban al azar)?
2. ¿Qué significa esto en términos del experimento de Emily? ¿Las respuestas de los practicantes de TT siguen un modelo de probabilidad equitativa o hay algo más en juego?

Pista: La respuesta corresponde a hacer un arreglo de 2 dimensiones, tipo `bool` (Verdadero o Falso)

In [46]:

```
# BEGIN SOLUTION NO PROMPT

peer_talk = np.array([True, False])

# END SOLUTION
""" # BEGIN PROMPT
peer_talk = ...
peer_talk
""" # END PROMPT
```

Out[46]:

```
' # BEGIN PROMPT\npeer_talk = ...\npeer_talk\n'
```

In [ ]:

```
grader.check("q10")
```

# ¡CASI LLEGAMOS!

Estrellita quiere hacerte saber que eres genial.

¡Terminaste con el laboratorio!

## Información importante sobre el envío:

- **Ejecuta todas las pruebas** y verifica que todas pasan
- **Guardar** desde el menú **Archivo**
- **\*\*Ejecute la celda final \*\***
- Luego, ve a este [enlace \(https://cienciadatosudea.github.io/001\\_SitioWebFundCienciaDatos/04entregas.html\)](https://cienciadatosudea.github.io/001_SitioWebFundCienciaDatos/04entregas.html) y envía el archivo ipynb a la tarea correspondiente. El nombre de esta tarea es "Lab XX Autograder", donde XX es el número de laboratorio: 01, 02, 03, etc.
- Si terminas temprano en el laboratorio regular, **registra tu asistencia con el instructor**.

**Es su responsabilidad asegurarse de que su trabajo esté guardado antes de ejecutar la última celda.**

Referencia : Este material esta basado en [Data 8: The Foundations of Data Science \(https://www.data8.org/\)](https://www.data8.org/) - University of California, Berkeley.

## Envío

Asegúrese de haber ejecutado todas las celdas de su cuaderno en orden antes de ejecutar la siguiente celda, para que todas las imágenes/gráficos aparezcan en el resultado. La siguiente celda generará un archivo zip para que lo envíe. **¡Guarde antes de exportar!**

## Submission

Make sure you have run all cells in your notebook in order before running the cell below, so that all images/graphs appear in the output. The cell below will generate a zip file for you to submit. **Please save before exporting!**

These are some submission instructions.

In [ ]:

```
# Save your notebook first, then run this cell to export your submission.  
grader.export(run_tests=True)
```