{desafío}
latam_

Probabilidad y Funciones_

Sesión Presencial 1

{desafío}
latam_

Alcances de la lectura asignada

- *Utilizar funciones para reutilizar código. (Principio D.R.Y)*
- *Convertir una formula matemática a una función en python*
- *Construir y utilizar funciones orientadas al análisis de datos.*
- *Optimizar funciones remplazandolas por funciones vectorizadas.*
- *Utilizar conceptos básicos de probabilidad.*
- *Generar segmentaciones de un `pd.DataFrame` en base a indexación y selección.*

Activación de conceptos

- La semana pasada aprendimos sobre estadística univariada y control de flujo.
- ¡Pongamos a prueba nuestros conocimientos!

A esta línea de código le faltan elementos para correr de forma correcta

```
name = "ignacio"
print("hola, mi nombre es" Name)
```

- `name` no es una cadena.
- A `print` le falta una coma.
- La variable no está definida.
- La variable debe ir en mayúscula.

Si tenemos las siguientes variables

```
var_1 = 5.4
var_2 = "Hola!"
```

¿Cuál sería el retorno de `var_1 == var_2`?

- `False`
- `var_1 = 'Hola!`
- `True`
- `SyntaxError: invalid syntax`

¿Cómo ingresamos una base de datos a nuestro ambiente de trabajo?

- Copiamos y pegamos la base de datos.
- Utilizamos `pd.csv()`
- `import pandas as pd`
- `import pandas as pd` y posteriormente `df = pd.read_csv()`

Si queremos obtener el promedio de `array`, ¿Cuál de las siguientes maneras es correcta?

```
array = [5, 4, 2, 5, 7, 3, 4, 2, 7, 8, 10]
```

- `np.mean(array)`
- `pd.mean(array)`
- `sum(array) / len(array)`
- Ninguna de las anteriores.

Queremos elevar al cuadrado los números pares de un array. ¿Cómo lo hacemos?

- Eliminamos los números impares del array y elevamos con ** 2 cada elemento.
- Elevamos al cuadrado el array y eliminamos los números pares.
- Generamos un loop que recorra cada elemento del array y evaluamos cada elemento si es par.
- Generamos un nuevo array con los elementos pares previamente elevados al cuadrado.

¿Cómo implementamos la solución anterior?

A

```
for (i in array) {
    if (i % 2) {
        print(i ** 2)
       }
    }
```

B

```
if i % 2 ==0
    for i in range(array):
        print i^2
```

C

```
for i in array:
    if i % 2 == 0:
        print(i **2)
```

Probabilidad

{desafío}
latam_

¿Qué se entiende por probabilidad?

- Son juicios sobre experimentos **aleatorios** que producen una serie de resultados.
- Generalmente entendemos probabilidad como **la repetición de un experimento en el largo plazo** (Laplace), a diferencia de la probabilidad como **juicio sobre la creencia en un suceso específico** (Bayes).
- Dos elementos:
  - Un evento específico A
  - Un espacio muestral Ω

$$\Pr(A) = \frac{\text{Número de elementos del evento } A}{\text{Posibles resultados en } \Omega}$$

In [ ]:
```
# con pandas utilizábamos el método value_counts() en una serie
df['major_death'].value_counts() / len(df)
```

Eventos Independientes

- Eventos donde la probabilidad de ocurrencia de A no se ve afectada por B.

Eventos Dependientes

- Eventos donde la probabilidad de A se ve afectada por B.

Intersección de Eventos

In [9]:
```
counter = np.logical_and(df['major_death'] == 1.0,
                         df['major_capture'] == 1.0)
np.unique(counter, return_counts=True)
```

Out[9]:
```
(array([False,  True]), array([32,  6]))
```

Unión de Eventos

In [11]:
```
counter = np.logical_or(df['major_death'] == 1.0,
                        df['major_capture'] == 1.0)

np.unique(counter, return_counts=True)
```

Out[11]:
```
(array([False,  True]), array([20, 18]))
```

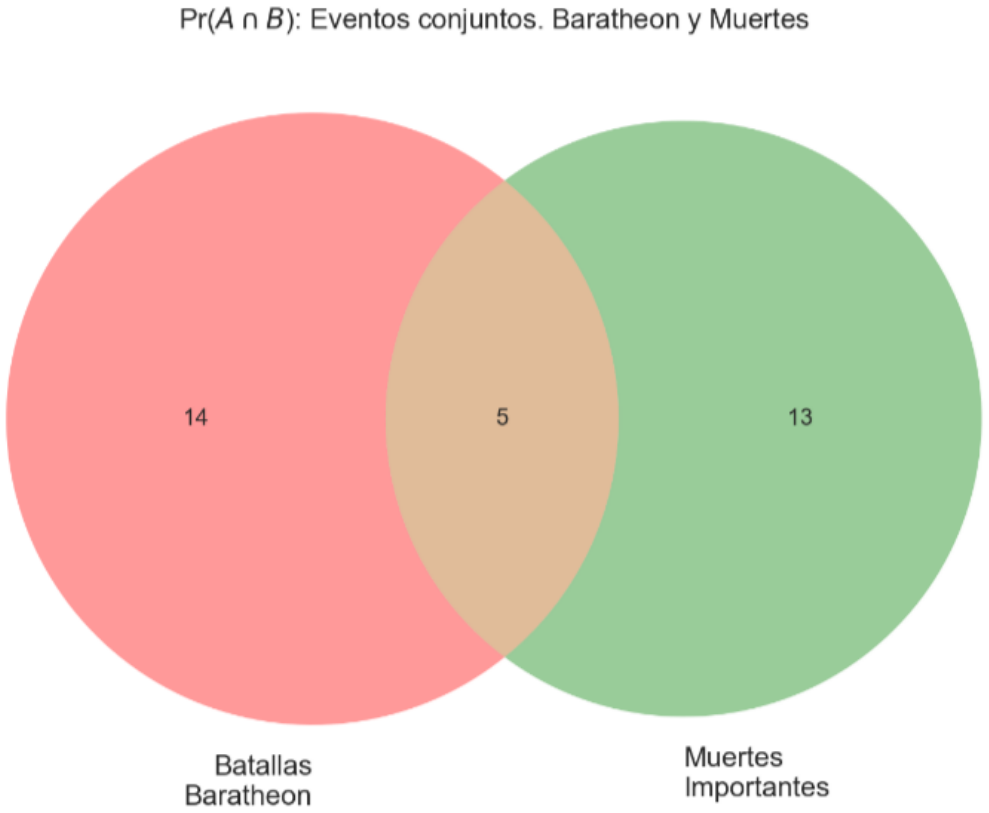
Probabilidad Condicional

- Busca responder a la verosimilitud de un evento A dado que B.
- Debemos buscar la probabilidad de $\Pr(A|B)$

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}$$

- Donde $\Pr(A \cap B)$ es la *intersección* de ambos eventos.

In [22]:
```
gfx.graph_venn()
```



Implementando la probabilidad condicional con Python

- Podemos dividir la tarea en tres pasos:

1. Contar las ocurrencias de A y B.

- Recorrer los datos y sumar cuando ambos eventos estén presentes
- Dividir la intersección por la cantidad de batallas

Contar las ocurrencias de A y B

In [ ]:
```
muerte_baratheon = 0
```

Recorrer los datos y sumar cuando ambos eventos estén presentes

- Utilizamos un loop con `iterrows`, que facilita separar entre filas y columnas.
- Generamos una expresión condicional donde ambos eventos existan.
- De ser así, sumar uno en nuestro contador.

In [ ]:
```
for filas, columnas in df.iterrows():
    if (r['attacker_king'] == 'Joffrey/Tommen Baratheon' and r['major_death'] == 1):
        muerte_baratheon += 1
```

Dividir la intersección por la cantidad de batallas

In [ ]:
```
batallas_baratheon = df['attacker_king'].value_counts()[0]
print("Pr(Muerte|Baratheon):", muerte_baratheon / batallas_baratheon)
```

In [ ]:
```
print("Cantidad de muertes importantes en batallas Baratheon: ", muerte_baratheon)
print("Pr(Muerte | Baratheon): ", muerte_baratheon / len(df))
```