



## Variables Aleatorias y Gráficos\_

Sesión Presencial 1

## Alcances de la unidad asignada

- Hacer uso de métodos de pandas para segmentar columnas y filas.
- Hacer uso de los métodos *iterrows* e *iteritems* para implementar loops en pandas.
- Implementar *enumerate* en loops.
- Conocer las convenciones y principios rectores de la visualización de gráficos.
- Conocer las principales convenciones en la visualización de resultados en histogramas, gráficos de punto y barras.
- Generar simulaciones de la distribución normal y
- Conocer las principales aplicaciones de las distribuciones.
- Calcular e interpretar puntajes  $z$
- Describir la Ley de los Grandes Números y Teorema del Límite Central y su importancia en la inferencia estadística.

## Activación de Conceptos

- En la unidad anterior aprendimos sobre probabilidades y flexibilizar nuestro flujo con funciones.
- ¡Pongamos a prueba nuestros conocimientos!

## ¿Cómo cargamos la librería de gráficos en Python?

---

- `from matplotlib import pyplot`
- `import matplotlib.pyplot`
- `Pkg.use("matplotlib")`
- `import matplotlib.pyplot as plt`

Si tenemos la siguiente tabla llamada df, ¿Cómo seleccionamos la cuarta columna?

---

i	a1	a2	a3
0	1	2	56
1	2	4	23
2	3	6	67
3	4	8	34
4	5	10	6
5	6	12	6

- `df[3]`
- `df.index(3)`
- `df.loc[3,:]`
- `df[:,3]`

Con la misma tabla, ¿Cómo seleccionamos la columna a2?

---

i	a1	a2	a3
0	1	2	56
1	2	4	23
2	3	6	67
3	4	8	34
4	5	10	6
5	6	12	6

- `df.loc[:, 'a2']`
- `df.slice['a2']`
- `df[3:,]`
- `df.column('a2')`

Con la misma tabla deseamos obtener la observación 56, ¿Cómo accedemos a ella?

---

i	a1	a2	a3
0	1	2	56
1	2	4	23
2	3	6	67
3	4	8	34
4	5	10	6
5	6	12	6

- `df.iloc[:, 'a2']`
- `df.loc[0, 'a2']`
- `df[3: 'a3']`
- `df.loc[0, 'a3']`

Queremos una función que al ingresar un nombre, nos salude. ¿Cuál de las siguientes es la correcta?

---

A

```
saludar <- function(name) {  
  print(paste("hola", name))  
}
```

B

```
def saludar(name):  
  print("hola", name)
```

C

```
def saludar(name)  
  print("Hola", name)  
end
```

D

```
def saludar(name)  
  print "hola", name
```



¿Cuál es el resultado de la siguiente función?

---

```
def funcion(x = 5):  
    for i in range(5):  
        print(i * 2)
```

funcion()

- Una lista con 0, 2, 4, 6, 8
- Nada, falta definir un argumento en la función.
- Una lista con 2, 4, 6, 8, 10

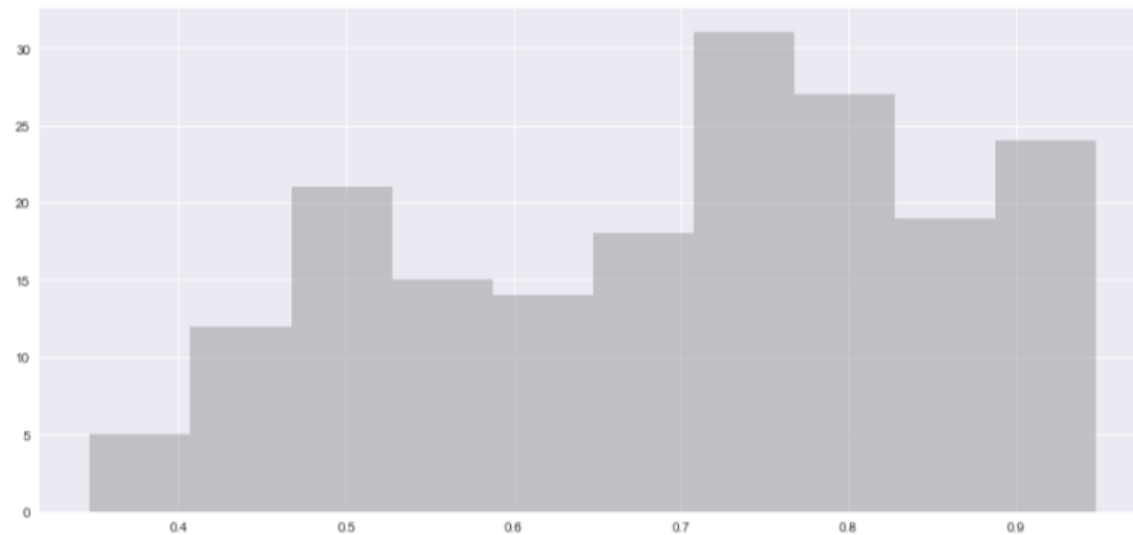
## Histogramas

{desafío}  
latam\_

## ¿Qué son y para qué sirven?

- Es la representación **empírica** de una variable.
- Permite visualizar cómo se comporta una variable. Por lo general se utilizan para las variables continuas.
- Se generan mediante `plt.hist`.
- Por defecto miden la frecuencia de casos entre cierto rango, que se llaman bins.

```
In [2]: # vamos a eliminar los datos perdidos en la columna undp_hdi con dropna()
hdi_dropna = df['undp_hdi'].dropna()
_,_ = plt.subplots(figsize = (15,7));
plt.hist(hdi_dropna, color='grey', alpha=.4);
```

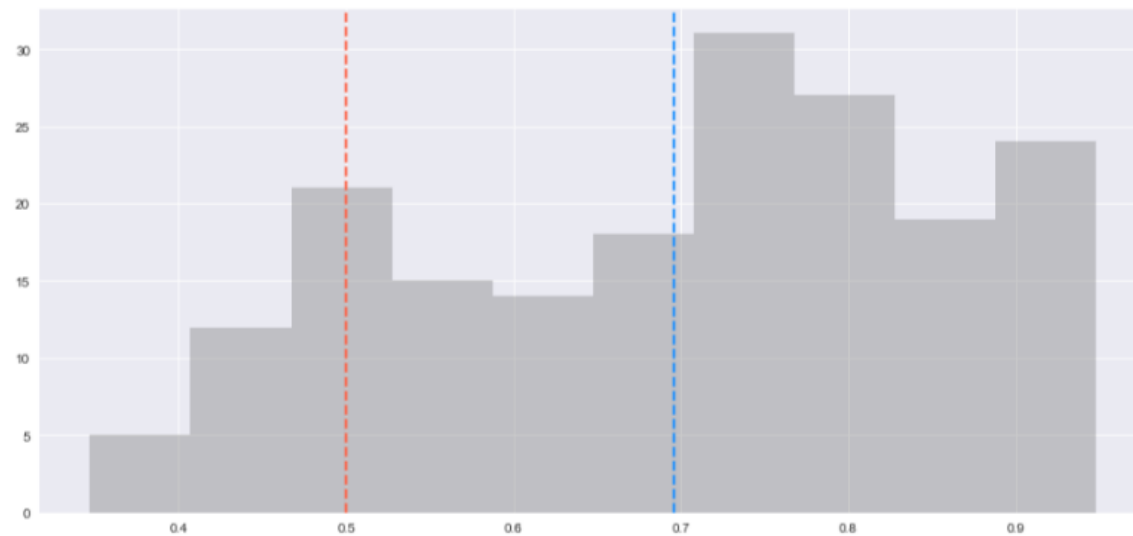


**{desafío}**  
latam\_

## matplotlib

- Al ejecutar el comando `plt.hist()`, realizamos nuestro primer gráfico utilizando `matplotlib.pyplot`.
- Con ello generamos implícitamente iniciamos un objeto *Figure*.
- Si realizamos más funciones `plt.` dentro de un mismo chunk, estaremos añadiendo o modificando información sobre el mismo gráfico.

```
In [3]: #guardemos la media en un objeto
hdi_mean = hdi_dropna.mean()
_,_ = plt.subplots(figsize = (15,7));
plt.hist(hdi_dropna, color='grey', alpha=.4)
plt.axvline(hdi_mean, color='dodgerblue', linestyle='--', lw=2)
plt.axvline(0.5, color='tomato', linestyle='--', lw=2);
```



## Variables Aleatorias Continuas

## ¿Qué son?

- Sirven para definir los posibles valores de una variable dado una distribución probabilística.
- Las variables aleatorias continuas son aquellas donde el fenómeno empírico se mide en un rango de valores continuos.
- La más conocida es la distribución normal, que representan fenómeno como altura, peso, y CI de una población:
  - Depende de la media y varianza.



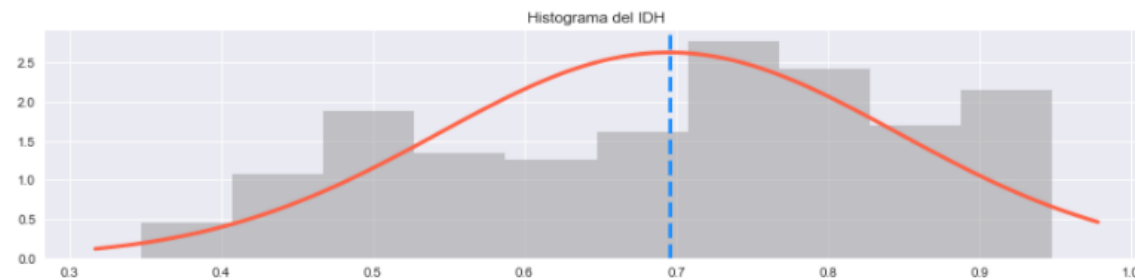
```

In [4]: _,_ = plt.subplots(figsize = (15,7))
plt.subplot(2,1,1)
plt.hist(hdi_dropna, color='grey', alpha=.4, normed=True)
x_min, x_max = plt.xlim()
x_axis = np.linspace(x_min, x_max, 100)
plt.plot(x_axis, stats.norm.pdf(x_axis, 0.695, np.sqrt(0.023)), color='tomato', lw=3)
plt.axvline(hdi_dropna.mean(), color='dodgerblue', linestyle='--', lw=3)
plt.title("Histograma del IDH")
...

/Users/veterok/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_
axes.py:6521: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be remove
d in 3.1. Use 'density' instead.
  alternative="'density'", removal="3.1")

```

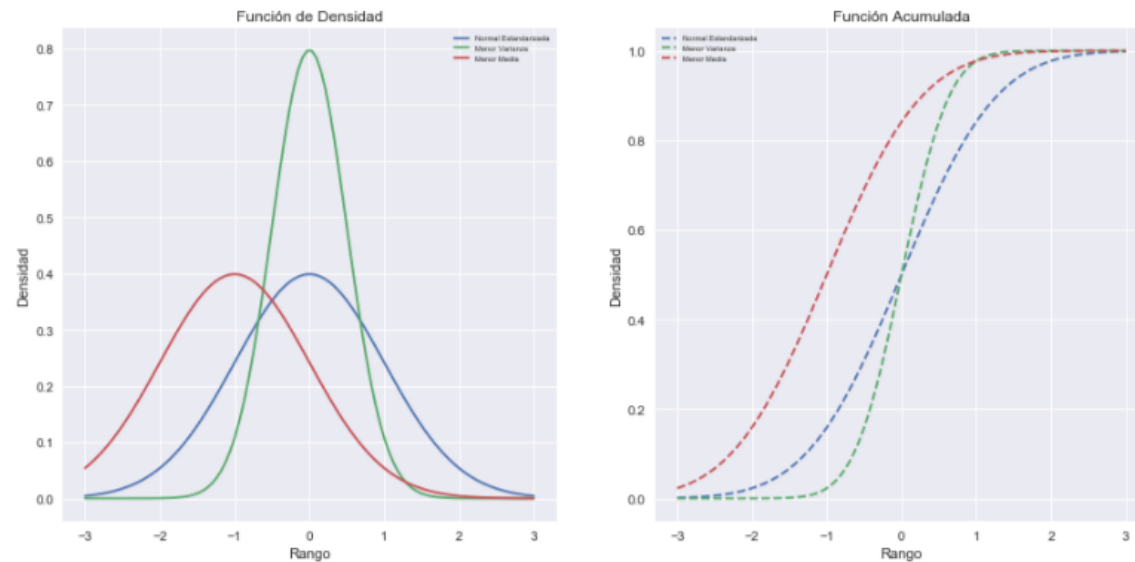
Out[4]: Ellipsis



**{desafío}**  
latam\_

## ¿Qué pasa al modificar los parámetros de nuestra distribución normal?

```
In [5]: _,_ = plt.subplots(figsize = (15,7))  
gfx.normal_distributions()
```



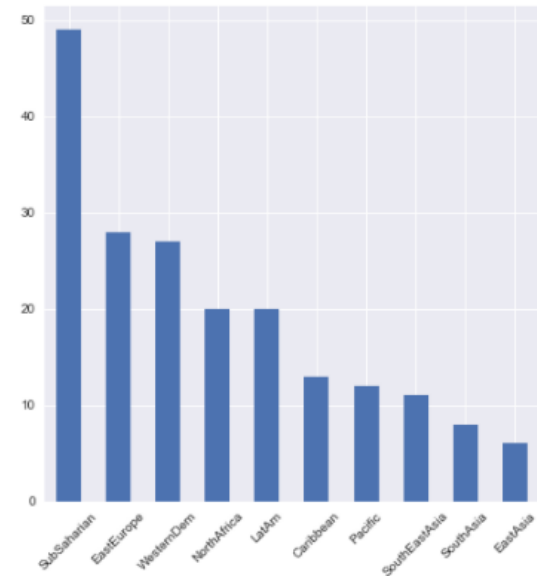
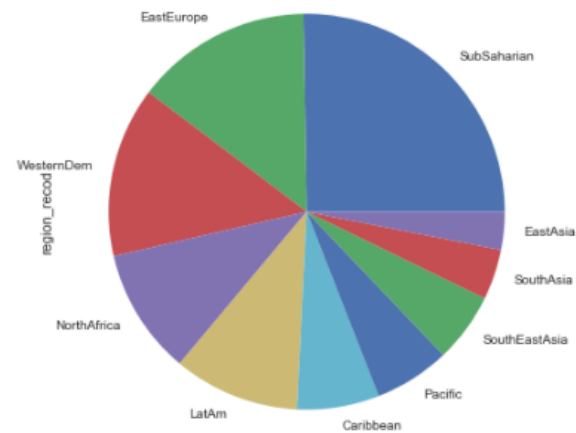
## Gráficos de Barras

## ¿Qué buscan representar?

- Muchas veces los números de nuestras variables representan categorías, por lo que su orden no es explícito y requiere ser definido por el investigador.
- Los gráficos de barra permiten evaluar de manera global cuál es la categoría dominante y facilitar la comparación entre categorías.

```
In [6]: _,_ = plt.subplots(figsize = (15,7))

plt.subplot(1,2,1)
hdi_group.region_recod.value_counts().plot(kind='pie')
plt.subplot(1,2,2)
hdi_group.region_recod.value_counts().plot(kind='bar')
plt.xticks(rotation = 45);
```



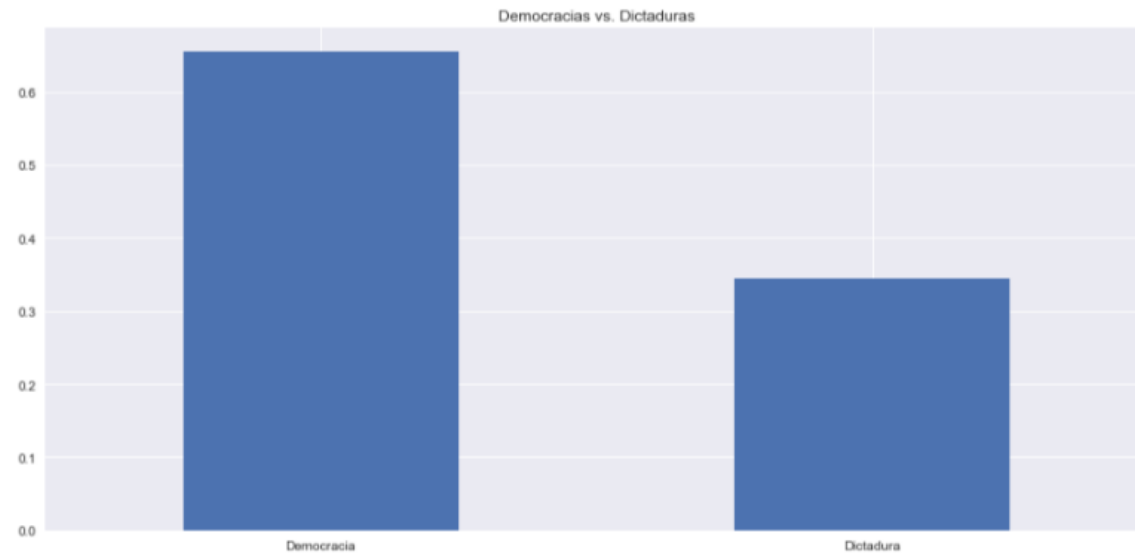
{desafío}  
latam\_

## Variables Aleatorias Discretas

## Aproximando el comportamiento de categorías

- Cuando realizamos un juicio probabilístico, utilizamos de forma implícita una distribución **Bernoulli**. Las probabilidades se representan como:
  - Éxito:  $p$
  - Fracaso:  $q = (1 - p)$
- Por si misma, **Bernoulli** hace referencia a *un solo evento*. Generalmente estamos interesados en la suma de una serie de ensayo Bernoulli. Esta suma se conoce como *variables binomiales*.

```
In [7]: _,_ = plt.subplots(figsize = (15,7))
df['democracies'] = np.where(df['gol_inst'] <= 2, 'Democracia', 'Dictadura')
df['democracies'].value_counts('%').plot(kind='bar')
plt.title('Democracias vs. Dictaduras')
plt.xticks(rotation = 0);
```



**{desafío}**  
**latam\_**



```
In [8]: mu, sigma = stats.binom(len(df), .654).stats()
print("Pr(Dem == 1): ", mu.round(2))
print("sigma(Dem == 1): ", sigma.round(2))
```

```
Pr(Dem == 1): 126.88
sigma(Dem == 1): 43.9
```

**{desafío}**  
**latam\_**

## Suma de ensayos Bernoulli

- Lanzar una moneda al azar y esperar que sea sello es un *ensayo de Bernoulli*.
- Si repetimos 10 veces éste experimento y sumamos la cantidad de sellos, obtendremos una variable binomial.
- Una variable Binomial depende de los parámetros  $n$  (cantidad de casos) y  $p$  (probabilidad del suceso).