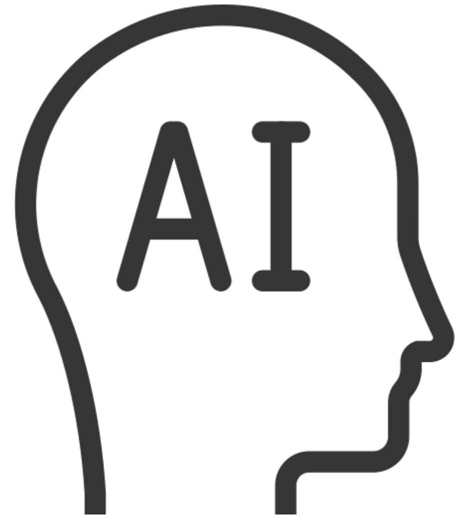
2.2 Distribución en ciencia de datos

Dr. Marco Aceves

Contenido

- Temario
  - Introducción.
  - Distribución de datos.
    - Tipos de distribución
  - Pruebas de hipótesis en ciencia de datos

Distribución de datos

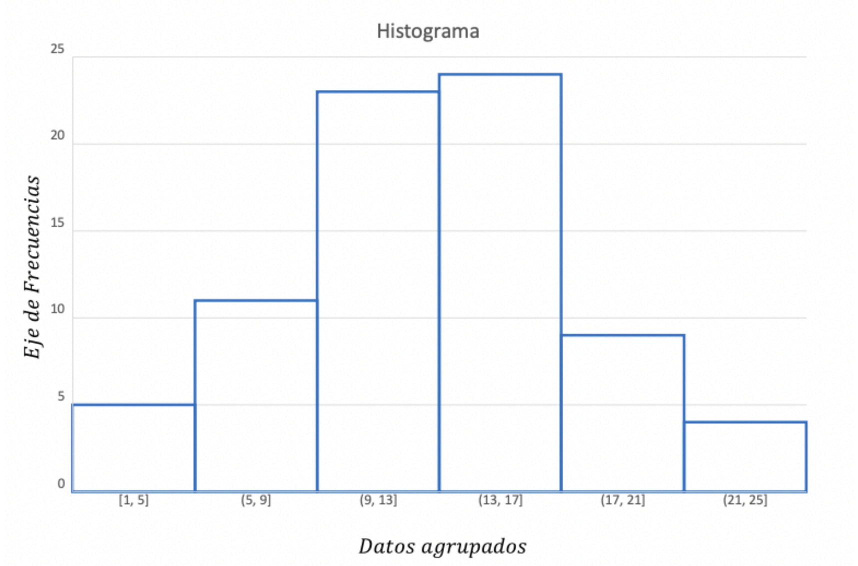


Distribución de datos

- Para manejo de datos, se tiene que examinar las características de los atributos. Para esto, se recomiendo primero examinar su media y desviación estándar. Esto se realiza para poder tener una idea de la tendencia central de los datos y la variación de los valores dentro del set de datos. También se tienen que evaluar los valores mínimos y máximos para entender el rango en los que se encuentra cada característica de los datos. Para ello existen los histogramas.

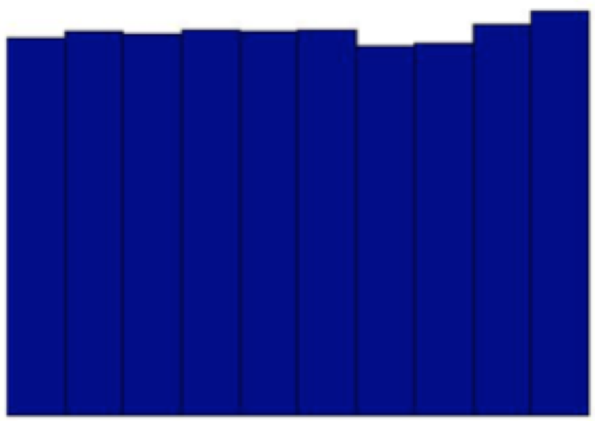
Histogramas

- Un histograma es una representación gráfica de la distribución de datos. El histograma está representado por un conjunto de rectángulos, adyacentes entre sí, donde cada barra representa un tipo de dato y su amplitud está directamente relacionada con la frecuencia del dato.

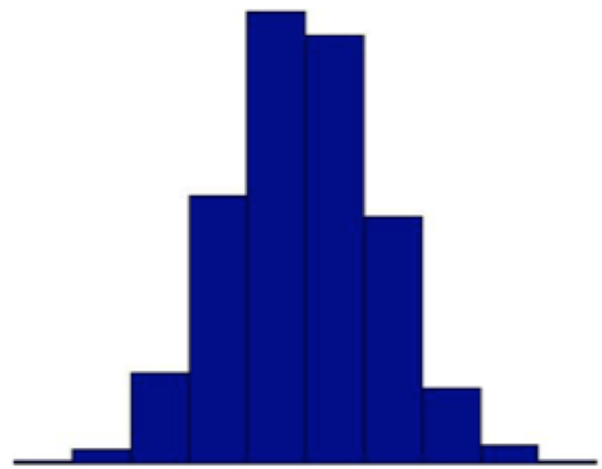


Distribución de datos

- Los histogramas de cada característica son útiles para entender cómo los valores de cada característica y cómo se distribuyen en un rango de datos.



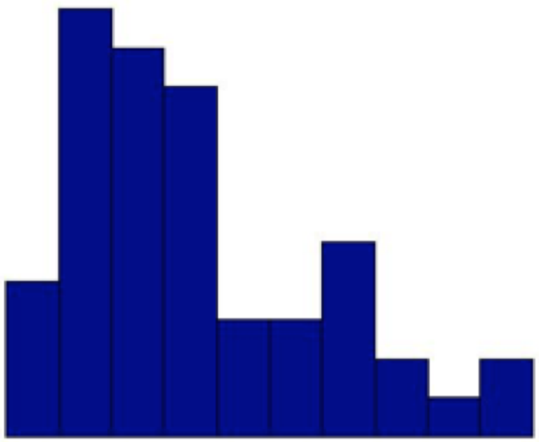
Uniforme



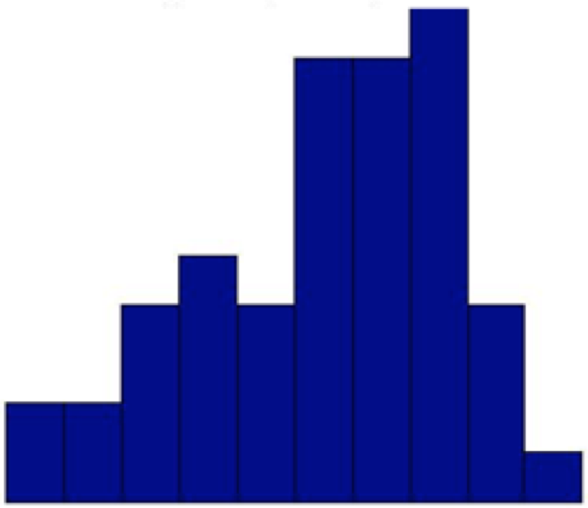
Normal (Unimodal)

Distribución de datos

- Los histogramas de cada característica son útiles para entender cómo los valores de cada característica y cómo se distribuyen en un rango de datos.



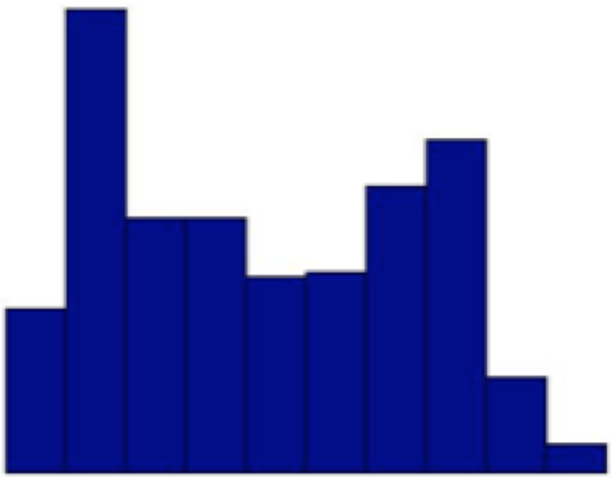
Unimodal (sesgado izquierda)



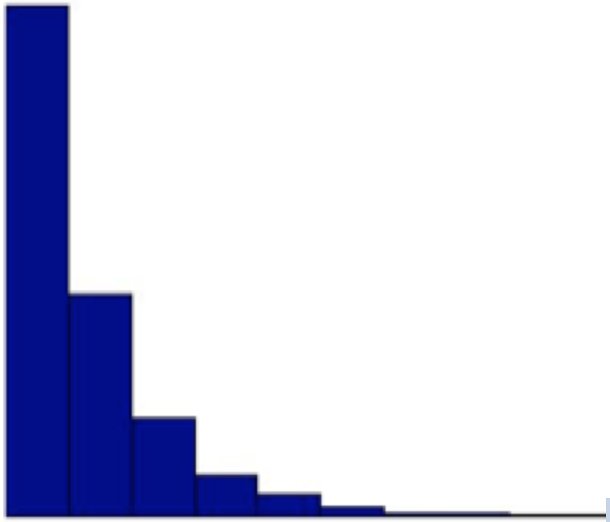
Unimodal (sesgado derecha)

Distribución de datos

- Los histogramas de cada característica son útiles para entender cómo los valores de cada característica y cómo se distribuyen en un rango de datos.



Multimodal



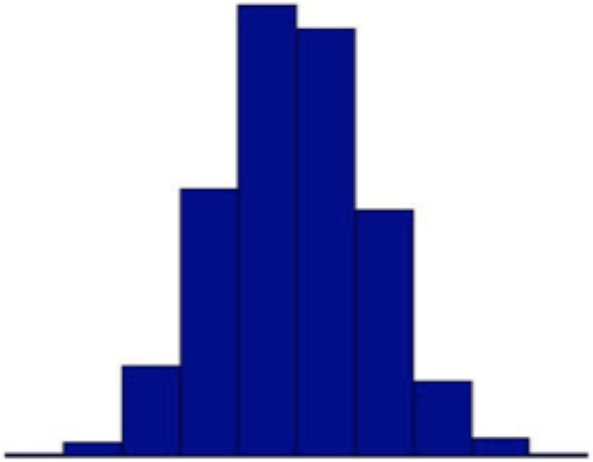
Exponencial

Distribución de datos

- Es importante mencionar que la simetría no tiene que ser perfecta para ser una distribución normal, pero si tener una medida de tendencia central. Como ejemplo, si se toman una cierta cantidad de datos al azar, por ejemplo, la altura de niños de 4to de primaria en alguna escuela. En este ejemplo, será muy probable que se tenga una distribución normal, con una cierta tendencia hacia una cierta estatura y a sus lados, niños y niñas mas altos y menos altos.

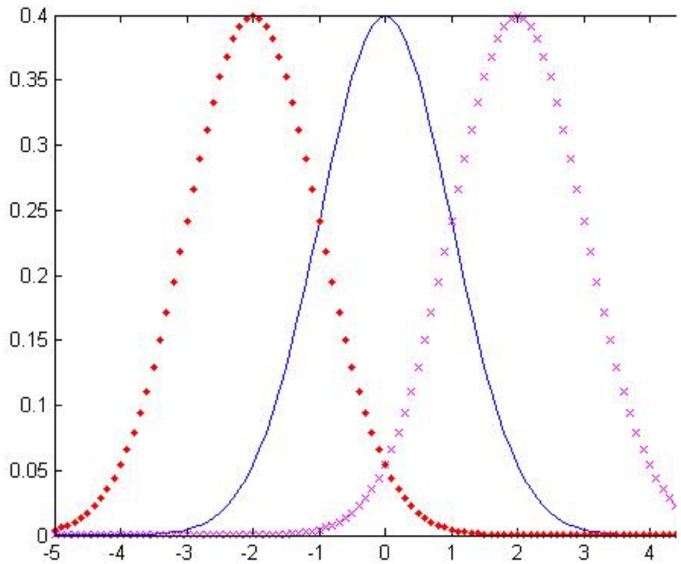
Normal (Unimodal)

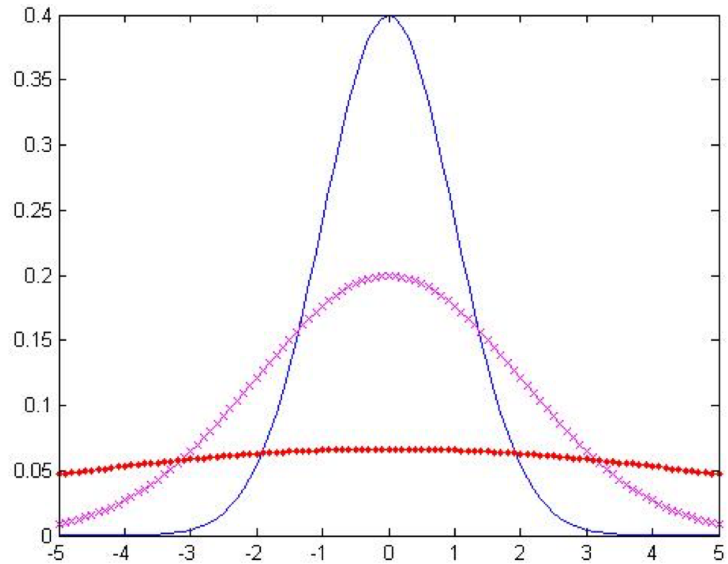
$$N(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



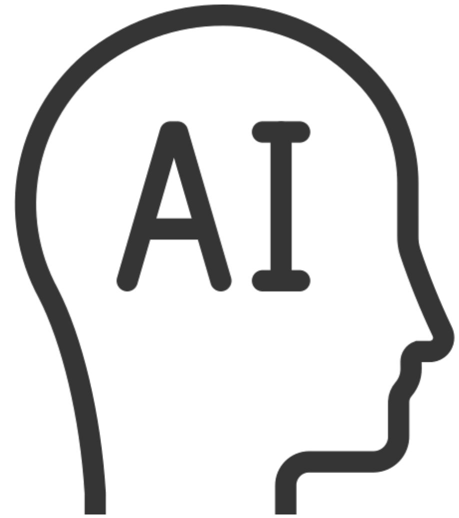
Distribución de datos

- Donde x es cualquier valor, y σ y μ son parámetros que definen la forma de la distribución Gaussiana. Es importante definir el tipo de la distribución, ya que la misma media pero diferente desviación estándar o viceversa puede generar una distribución muy diferente.





Prueba de hipótesis en ciencia de datos



Pruebas estadísticas de hipótesis

- En esta sección, se presentan algunas de las pruebas de hipótesis estadísticas más populares para un proyecto de aprendizaje automático, con ejemplos que utilizan la API de Python.
- Ten en cuenta que en lo que respecta a suposiciones como la distribución esperada de los datos o el tamaño de la muestra, es probable que los resultados de una prueba determinada se degraden de forma gradual en lugar de volverse inmediatamente inutilizables si se viola una suposición.
- En general, las muestras de datos deben ser representativas del dominio y lo suficientemente grandes como para exponer su distribución al análisis.
- En algunos casos, los datos se pueden corregir para cumplir con los supuestos como corregir una distribución casi normal para que sea normal eliminando valores atípicos, o utilizando una corrección a los grados de libertad en una prueba estadística cuando las muestras tienen varianzas diferentes, por mencionar dos ejemplos.

Pruebas estadísticas de hipótesis

- Cada prueba estadística se presenta de manera consistente, incluyendo:
  - El nombre de la prueba.
  - Lo que la prueba está verificando.
  - Las suposiciones clave de la prueba.
  - Cómo se interpreta el resultado de la prueba.
  - Código de python para usar la prueba.

Pruebas estadísticas de hipótesis

- Las pruebas que se abordarán en este capítulo son:
  - Pruebas de normalidad
    - 1.- Prueba de Shapiro-Wilk
    - 2.- Prueba de D'Agostino K^2
    - 3.- Prueba de Anderson-Darling
  - Pruebas de correlación
    - 4.- Coeficiente de correlación de Pearson
    - 5.- Correlación de rangos de Spearman
    - 6.- Correlación de rangos de Kendall
    - 7.- Prueba de Chi-cuadrado

Pruebas estadísticas de hipótesis

- Las pruebas que se abordarán en este capítulo son:
  - Pruebas de estacionariedad
    - 8.- Dickey-Fuller aumentado
    - 9.- Kwiatkowski-Phillips-Schmidt-Shin
  - Pruebas de hipótesis estadísticas paramétricas
    - 10.-Prueba t de Student
    - 11.- Prueba t de Student emparejada
    - 12.- Prueba de análisis de varianza (ANOVA)

Pruebas estadísticas de hipótesis

- Las pruebas que se abordarán en este capítulo son:
  - Pruebas de hipótesis estadísticas no paramétricas
    - 13.- Prueba U de Mann-Whitney
    - 14.- Prueba de rangos con signo de Wilcoxon
    - 15.- Prueba H de Kruskal-Wallis
    - 16.- Prueba de Friedman

Pruebas estadísticas de hipótesis

- Pruebas de normalidad
  - Esta sección enumera pruebas estadísticas que puedes utilizar para comprobar si tus datos tienen una distribución Gaussiana.
- **1.- Prueba de Shapiro-Wilk**
- Prueba si una muestra de datos tiene una distribución Gaussiana.

- *Supuestos*
- Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid).

- *Interpretación*
- *H0:* la muestra tiene una distribución Gaussiana.
- *H1:* la muestra no tiene una distribución Gaussiana.

Pruebas estadísticas de hipótesis

- **Prueba de Shapiro-Wilk**
- Código:
- from scipy.stats import shapiro
- data = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- stat, p = shapiro(data)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente Gaussiana')
- else:
- print('Probablemente no Gaussiana’)

```
stat=0.895, p=0.193
Probablemente Gaussiana
```

Pruebas estadísticas de hipótesis

- **2.- Prueba de K^2 de D'Agostino's**
- Prueba si una muestra de datos tiene una distribución Gaussiana.

- *Supuestos*
- Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid).

- *Interpretación*
- *H0:* la muestra tiene una distribución Gaussiana.
- *H1:* la muestra no tiene una distribución Gaussiana.

Pruebas estadísticas de hipótesis

- **2.- Prueba de K^2 de D'Agostino's**
- Código:
- from scipy.stats import normaltest
- data = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- stat, p = normaltest(data)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente Gaussiana')
- else:
- print('Probablemente no Gaussiana')

```
stat=3.392, p=0.183
Probablemente Gaussiana
```

- **3.- Prueba de Anderson-Darling**
- Prueba si una muestra de datos tiene una distribución Gaussiana.

- *Supuestos*
- Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid).

- *Interpretación*
- *H0:* la muestra tiene una distribución Gaussiana.
- *H1:* la muestra no tiene una distribución Gaussiana.

Pruebas estadísticas de hipótesis

- **3.- Prueba de Anderson-Darling**
- Código:
- from scipy.stats import anderson
- data = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- result = anderson(data)
- print('stat=%.3f' % (result.statistic))
- for i in range(len(result.critical_values)):
- sl, cv = result.significance_level[i], result.critical_values[i]
- if result.statistic < cv:
-     print('Probablemente Gaussiana al %.1f%% ' % (sl))
- else:
-     print('Probablemente no Gaussiana al %.1f%% ' % (sl))

```
stat=0.424
Probablemente Gaussiana al 15.0%
Probablemente Gaussiana al 10.0%
Probablemente Gaussiana al 5.0%
Probablemente Gaussiana al 2.5%
Probablemente Gaussiana al 1.0%
```

Pruebas estadísticas de hipótesis

- **4.- Prueba de correlación de Pearson**
- Prueba si dos muestras tienen una relación lineal.
- Supuestos:

- Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid).
- Las observaciones en cada muestra siguen una distribución normal.
- Las observaciones en cada muestra tienen la misma varianza.

- *Interpretación:*
- *H0: las dos muestras son independientes.*
- *H1: hay una dependencia entre las muestras.*

Pruebas estadísticas de hipótesis

- **4.- Prueba de correlación de Pearson**
- Código:
- from scipy.stats import pearsonr
- data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
- stat, p = pearsonr(data1, data2)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente independiente')
- else:
- print('Probablemente dependiente')

```
stat=0.688, p=0.028
Probablemente dependiente
```

Pruebas estadísticas de hipótesis

- **5.- Correlación de rango de Spearman's**
- Prueba si dos muestras tienen una relación monotónica.

- Supuestos:
- Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid).
- Las observaciones en cada muestra pueden ser clasificadas por rango.

- *Interpretación:*
- *H0: las dos muestras son independientes.*
- *H1: existe una dependencia entre las muestras.*

- **5.- Correlación de rango de Spearman's**
- Código:

```
from scipy.stats import spearmanr
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
stat, p = spearmanr(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probablemente independiente')
else:
    print('Probablemente dependiente')
```

```
stat=0.855, p=0.002
Probablemente dependiente
```

Pruebas estadísticas de hipótesis

- **6.- Correlación de rangos de Kendall**
- Prueba si dos muestras tienen una relación monotónica.

- Supuestos:
- Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid).
- Las observaciones en cada muestra pueden ser clasificadas por rango.

- *Interpretación:*
- *H0: las dos muestras son independientes.*
- *H1: existe una dependencia entre las muestras.*

Pruebas estadísticas de hipótesis

- **6.- Correlación de rangos de Kendall**
- Código:

```
from scipy.stats import kendalltau
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
stat, p = kendalltau(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=0.733, p=0.002
Probablemente dependiente
```

Pruebas estadísticas de hipótesis

- **7.- Prueba Chi-Cuadrado**
- Prueba si dos variables categóricas están relacionadas o son independientes.

- Supuestos:
- Las observaciones utilizadas en el cálculo de la tabla de contingencia son independientes.
- Hay 25 o más ejemplos en cada celda de la tabla de contingencia.

- *Interpretación:*
- *H0: las dos muestras son independientes.*
- *H1: existe una dependencia entre las muestras.*

- **7.- Prueba Chi-Cuadrado**
- Código:
- from scipy.stats import chi2_contingency
- table = [[10, 20, 30],[6, 9, 17]]
- stat, p, dof, expected = chi2_contingency(table)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente independiente')
- else:
- print('Probablemente dependiente')

```
stat=0.272, p=0.873
Probablemente independiente
```

Pruebas estadísticas de hipótesis

- **Pruebas estacionarias**
- Esta sección enumera pruebas estadísticas que se pueden utilizar para verificar si una serie temporal es estacionaria o no.
- **8.- Prueba de raíz unitaria aumentada Dickey-Fuller**
- Prueba si una serie temporal tiene una raíz unitaria, es decir, si tiene una tendencia o más generalmente es autorregresiva.
- Supuestos
    - Las observaciones están ordenadas temporalmente.
- *Interpretación:*
- H0: Hay una raíz unitaria presente (la serie no es estacionaria).
- H1: No hay una raíz unitaria presente (la serie es estacionaria).

- **9.- Prueba de raíz unitaria aumentada Dickey-Fuller**
- Prueba si una serie temporal tiene una raíz unitaria, es decir, si tiene una tendencia o más generalmente es autorregresiva.
- Supuestos
  - Las observaciones están ordenadas temporalmente.
- *Interpretación:*
- H0: Hay una raíz unitaria presente (la serie no es estacionaria).
- H1: No hay una raíz unitaria presente (la serie es estacionaria).

Pruebas estadísticas de hipótesis

- **9.- Prueba de raíz unitaria aumentada Dickey-Fuller**
- Código:
- from statsmodels.tsa.stattools import kpss
- data = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- stat, p, lags, crit = kpss(data)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente estacionaria')
- else:
- print('Probablemente no estacionaria')

```
stat=0.594, p=0.023
Probablemente no estacionaria
```

Pruebas estadísticas de hipótesis

- **Pruebas de hipótesis estadísticas paramétricas**
- Esta sección lista pruebas estadísticas que se pueden utilizar para comparar muestras de datos.
- **10 - Prueba t de Student**
- Prueba si las medias de dos muestras independientes son significativamente diferentes.
- Supuestos:
  - Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid).
  - Las observaciones en cada muestra tienen una distribución normal.
  - Las observaciones en cada muestra tienen la misma varianza.
- Interpretación
  - H0: las medias de las muestras son iguales.
  - H1: las medias de las muestras son diferentes.

Pruebas estadísticas de hipótesis

- **10 - Prueba t de Student**
- Código:
- from scipy.stats import ttest_ind
- data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
- stat, p = ttest_ind(data1, data2)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print(‘Probablemente la misma distribución')
- else:
- print(‘Probablemente diferente distribución')

```
stat=-0.326, p=0.748
Probablemente la misma distribución
```

Pruebas estadísticas de hipótesis

- **11 - Prueba t de Student emparejada**
- Prueba si las medias de dos muestras emparejadas son significativamente diferentes.
- Supuestos
  - Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid).
  - Las observaciones en cada muestra siguen una distribución normal.
  - Las observaciones en cada muestra tienen la misma varianza.
  - Las observaciones entre las muestras están emparejadas.
- *Interpretación*
  - H0: las medias de las muestras son iguales.
  - H1: las medias de las muestras son diferentes.

Pruebas estadísticas de hipótesis

- **11 - Prueba t de Student emparejada**
- Código:
- from scipy.stats import ttest_rel
- data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
- stat, p = ttest_rel(data1, data2)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente la misma distribución')
- else:
- print('Probablemente diferente distribución')

```
stat=-0.334, p=0.746
Probablemente la misma distribución
```

Pruebas estadísticas de hipótesis

- **12. Prueba de Análisis de Varianza (ANOVA)**
- Esta prueba determina si las medias de dos o más muestras independientes son significativamente diferentes.
- Suposiciones:
  - Las observaciones en cada muestra son independientes y distribuidas de manera idéntica (iid, por sus siglas en inglés).
  - Las observaciones en cada muestra siguen una distribución normal.
  - Las observaciones en cada muestra tienen la misma varianza.
- Interpretación:
  - H0: las medias de las muestras son iguales.
  - H1: una o más de las medias de las muestras son diferentes.

Pruebas estadísticas de hipótesis

- **12. Prueba de Análisis de Varianza (ANOVA)**
- Código:
- from scipy.stats import f_oneway
- data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
- data3 = [-0.208, 0.696, 0.928, -1.148, -0.213, 0.229, 0.137, 0.269, -0.870, -1.204]
- stat, p = f_oneway(data1, data2, data3)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente la misma distribución')
- else:
- print('Probablemente diferente distribución')

```
stat=0.096, p=0.908
Probablemente la misma distribución
```

Pruebas estadísticas de hipótesis

- **Pruebas de hipótesis estadísticas no paramétricas**

- **13. Prueba de U de Mann-Whitney**
- Prueba si las distribuciones de dos muestras independientes son iguales o no.
- *Suposiciones*
  - Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid).
  - Las observaciones en cada muestra pueden ser clasificadas.
- Interpretación:
  - H0: las distribuciones de ambas muestras son iguales.
  - H1: las distribuciones de ambas muestras no son iguales.

Pruebas estadísticas de hipótesis

- **13. Prueba de U de Mann-Whitney**
- Código:
- from scipy.stats import mannwhitneyu
- data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
- stat, p = mannwhitneyu(data1, data2)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente es la misma distribución')
- else:
- print('Probablemente es diferente distribución')

```
stat=40.000, p=0.473
Probablemente es la misma distribución
```

- **14. Prueba de rangos con signo de Wilcoxon**
- Prueba si las distribuciones de dos muestras emparejadas son iguales o no.
- *Suposiciones:*
  - Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid, por sus siglas en inglés).
  - Las observaciones en cada muestra pueden ser ordenadas por rangos.
  - Las observaciones entre las muestras están emparejadas.
- *Interpretación:*
  - H0: las distribuciones de ambas muestras son iguales.
  - H1: las distribuciones de ambas muestras no son iguales.

Pruebas estadísticas de hipótesis

- **14. Prueba de rangos con signo de Wilcoxon**
- Código:
- from scipy.stats import wilcoxon
- data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
- stat, p = wilcoxon(data1, data2)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente es la misma distribución')
- else:
- print('Probablemente es diferente distribución')

```
stat=21.000, p=0.557
Probablemente es la misma distribución
```

- **15. Prueba H de Kruskal-Wallis**
- Prueba si las distribuciones de dos o más muestras independientes son iguales o no.
- *Suposiciones:*
  - Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid).
  - Las observaciones en cada muestra pueden ser clasificadas.
- *Interpretación:*
  - H0: las distribuciones de todas las muestras son iguales.
  - H1: las distribuciones de una o más muestras no son iguales.

- **15. Prueba H de Kruskal-Wallis**
- Código:
- from scipy.stats import kruskal
- data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
- stat, p = kruskal(data1, data2)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente es la misma distribución')
- else:
- print('Probablemente es diferente distribución')

```
stat=0.571, p=0.450
Probablemente es la misma distribución
```

- **16. Prueba de Friedman**
- Prueba si las distribuciones de dos o más muestras emparejadas son iguales o no.
- *Suposiciones:*
  - Las observaciones en cada muestra son independientes e idénticamente distribuidas (iid, por sus siglas en inglés).
  - Las observaciones en cada muestra pueden ser clasificadas o ordenadas.
  - Las observaciones entre cada muestra están emparejadas.
- Interpretación:
  - H0: las distribuciones de todas las muestras son iguales.
  - H1: las distribuciones de una o más muestras no son iguales.

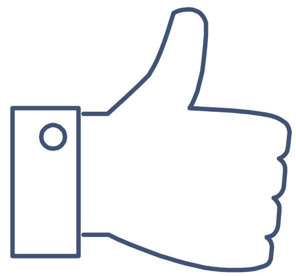
Pruebas estadísticas de hipótesis

- **16. Prueba de Friedman**
- Código:
- from scipy.stats import friedmanchisquare
- data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
- data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
- data3 = [-0.208, 0.696, 0.928, -1.148, -0.213, 0.229, 0.137, 0.269, -0.870, -1.204]
- stat, p = friedmanchisquare(data1, data2, data3)
- print('stat=%.3f, p=%.3f' % (stat, p))
- if p > 0.05:
- print('Probablemente es la misma distribución')
- else:
- print('Probablemente es diferente distribución’)

```
stat=0.800, p=0.670
Probablemente es la misma distribución
```

Pruebas estadísticas de hipótesis

- Práctica:
- Para cada prueba, realizar lo siguiente:
  - Un ejemplo de cada uno
  - Explicar en que consiste la prueba
  - Mostrar ecuaciones
  - Mostrar gráficas de los datos y/o pruebas
  - Casos de uso (es decir, cuándo se usa la prueba en cuestión)
  - Hacer un reporte y entregar la libreta pertinente



Gracias por tu atención

Cualquier pregunta, me la puedes hacer a:
marco.aceves@gmail.com