

ACÀMICA

¡Bienvenidos/as a Data Science!



Agenda

¿Cómo anduvieron?

Repaso: Probabilidad y Estadística

Explicación (con Hands-On): Correlación

Break

Explicación (con Hands-On): Visualización de Datos

¿Sabías que...?

Cierre



¿Cómo anduvieron?



Repaso: Probabilidad y Estadística



Probabilidad: Variables aleatorias

X variable aleatoria. Posibles resultados de un proceso aleatorio:

$X_{\text{moneda}} : \{\text{cara, ceca}\}$

$X_{\text{dado}} : \{1,2,3,4,5,6\}$

$X_{\text{clima}} : \{\text{lluvia, no lluvia}\}$

$X_{\text{clima}} : \{\text{cuánto llovió}\}$

$X_{\text{avión}} : \{\text{accidente, no-accidente}\}$



Probabilidad: Variables aleatorias

PROBABILIDAD



Variables **discretas**

- Son aquellas que se *cuentan*
- Pueden estar acotadas o no

Ejemplo: edades (en años), número de hijos, cantidad de dormitorios en una casa, etc.

Variables **continuas**

- Son aquellas que se *miden*
- Pueden estar acotadas o no

Ejemplo: altura de una persona, temperaturas, edades (medidas en tiempo transcurrido desde el nacimiento), etc.

Variables discretas: Distribución

La distribución de probabilidad de una variable aleatoria es una función que asigna a cada suceso definido sobre la variable la probabilidad de que dicho suceso ocurra.



Para variables **continuas** se usa el concepto de **densidad** de probabilidad.



Probabilidad: Densidad normal o Gaussiana

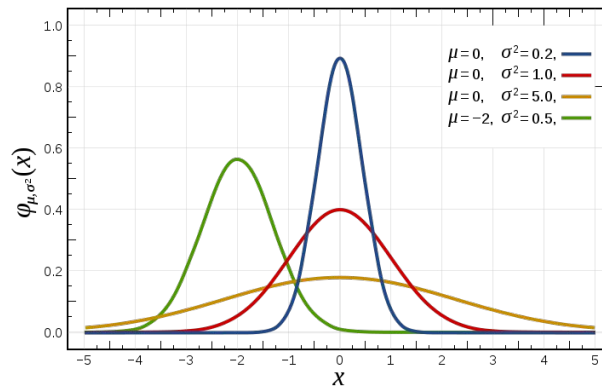
¡La más famosa de las distribuciones!

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

Parámetros:

μ : valor medio

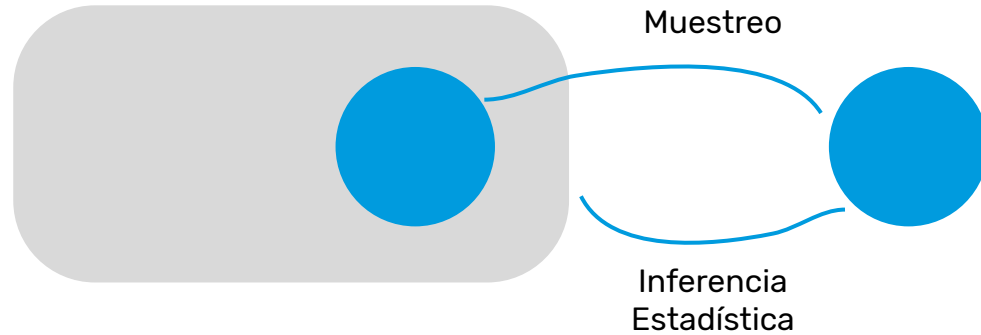
σ : desviación estándar



Población y Muestra

POBLACIÓN
(Parámetros)

MUESTRA
(Estadísticos)



+ Probabilidad y Estadística



Hasta ahora, consideramos **una sola variable aleatoria $X...$**

- Resultados obtenidos al tirar una moneda
- Resultados obtenidos al tirar un dado
- Altura de una persona
- Etc.

Llamamos probabilidad de X ,
 $P(X)$ a un número entre 0 y 1
que le asignamos a cada
posible valor que puede
tomar esa variable aleatoria

Llamamos probabilidad de X ,
 $P(X)$ a un número entre 0 y 1
que le asignamos a cada
posible valor que puede
tomar esa variable aleatoria

También dijimos que la **mejor descripción** de $P(X)$ está dada por la distribución **(densidad) de probabilidades.**

¿Pero qué ocurre si tengo más de una variable aleatoria?

¿Pero qué ocurre si tengo más de una variable aleatoria?

2 variables

Peso y altura de una persona
Temperatura y humedad

3 variables

Temperatura, humedad y velocidad del viento

4 variables

Iris Dataset: ancho y largo del sépalos y pétalo

Podemos pensar a un
Dataset como
una “colección” de
variables aleatorias

Probabilidad Conjunta

Dadas dos variables aleatorias X e Y

- **$P(X)$** es la distribución (o densidad) de probabilidades de X
- **$P(Y)$** es la distribución (o densidad) de probabilidades de Y
- **$P(X,Y)$** es la probabilidad CONJUNTA de X y de Y

Probabilidad Conjunta

$P(X,Y)$ es la probabilidad CONJUNTA de X y de Y

Es la distribución de probabilidad de los pares (x,y) , es decir, todos los posibles valores que pueden tomar las dos variables.

Probabilidad Conjunta

$P(X,Y)$ es la probabilidad CONJUNTA de X y de Y

Es la distribución de probabilidad de los pares (x,y) , es decir, todos los posibles valores que pueden tomar las dos variables.



Probabilidad Conjunta

$P(X,Y)$ es la probabilidad CONJUNTA de X y de Y

¡Veamos un ejemplo!

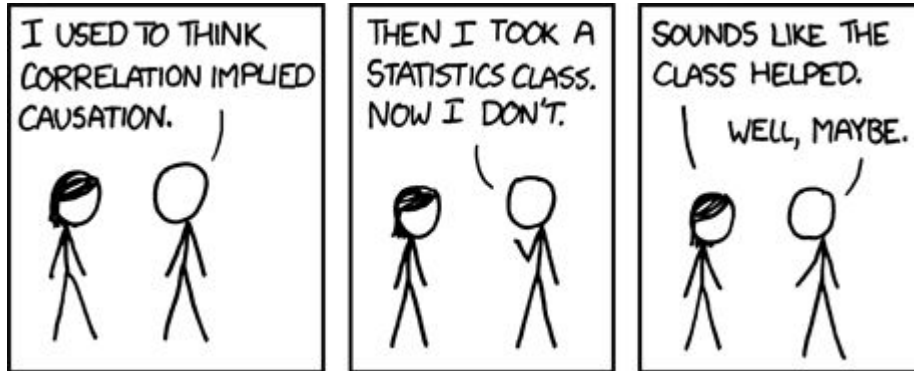
Medimos para muchas personas su peso y altura. Cada par (peso, altura) es una “muestra” de la distribución conjunta $P(X,Y)$.

Correlación, Causalidad e Independencia estadística

- Tres conceptos que tratan sobre la relación entre dos variables aleatorias.
- Muy fácil confundirlos entre ellos
- En su uso cotidiano tienen un significado un poco más “laxo” que en su uso estadístico

Correlación, Causalidad e Independencia estadística

Hoy hablaremos de correlación
(y un poco de Causalidad)



¡No podía faltar!

Hands-on training



DS_Clase_06_Correlacion.ipynb



Conclusiones

- Correlación no implica causalidad
- La correlación de Pearson es muy útil para encontrar correlaciones lineales
- Si la relación entre las variables NO es lineal, existen otras correlaciones que pueden ser útiles: Spearman y Kendall

Recursos



Links

[Correlaciones Espurias](#)

Visita obligada y muy divertida.

[Statistics is the Grammar of Data Science](#)

Parte cuatro de una serie de cinco sobre PyE.



A close-up photograph of a white ceramic cup filled with a latte. The surface of the milk is decorated with intricate latte art, featuring a central heart shape surrounded by concentric, wavy lines. The cup is placed on a matching white saucer. In the background, a white napkin and a silver fork are visible, though they are out of focus. The overall lighting is soft and even, highlighting the textures of the coffee and the smooth surface of the cup.

¡BREAK!



Visualización de Datos



¿Por qué visualizar?

Visualizar los datos es una parte fundamental del análisis en ciencia de datos.



¿Por qué visualizar?

No solo sirve para **comunicar** (que es una parte fundamental del trabajo) sino que también es una herramienta esencial para **comprender el dataset** con el que estamos trabajando.

¿Por qué visualizar?

Hay veces que sólo indicadores numéricos no alcanzan para describir las características principales de nuestro dataset.

Supongamos un par de variables (x,y)
cuyo **coeficiente de correlación** de
Pearson es igual a cero.
¿Cómo imaginan su distribución?

¿Por qué visualizar?

Hay veces que sólo indicadores numéricos no alcanzan para describir las características principales de nuestro dataset.

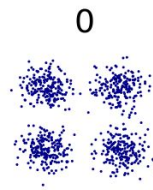
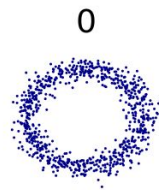
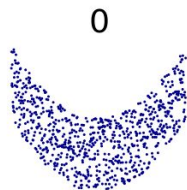
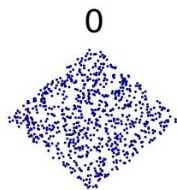
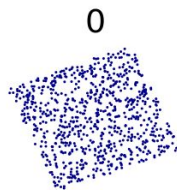
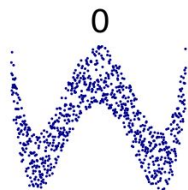
Supongamos un par de variables (x,y) cuyo **coeficiente de correlación** de Pearson es igual a cero.
¿Cómo imaginan su distribución?



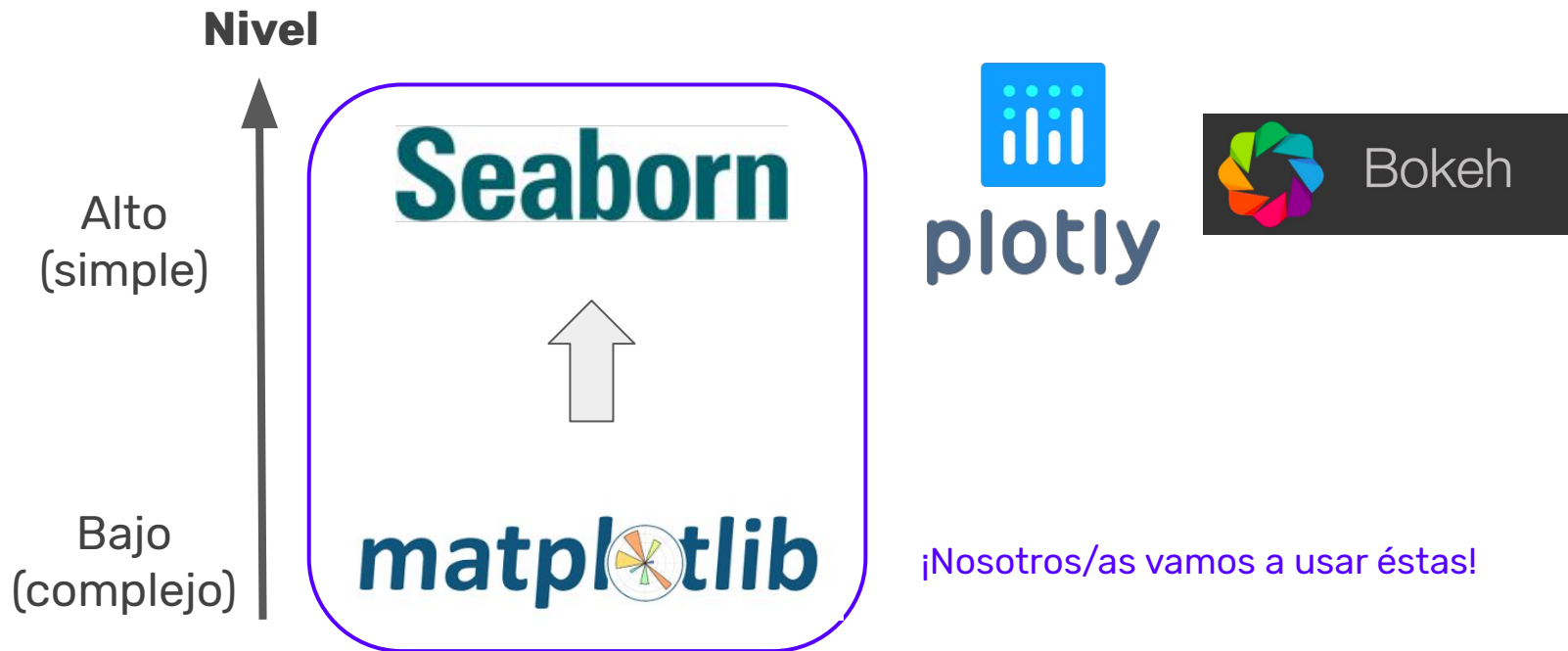
¿Por qué visualizar?

Hay veces que sólo indicadores numéricos no alcanzan para describir las características principales de nuestro dataset.

Supongamos un par de variables (x,y) cuyo **coeficiente de correlación** de Pearson es igual a cero.
¿Cómo imaginan su distribución?



Herramientas de Visualización



Introducción a Matplotlib

Por si no la tienen instalada de clases anteriores:

conda install matplotlib

La documentación de matplotlib es **excelente**, es importantísimo aprovecharla:

<https://matplotlib.org/index.html>

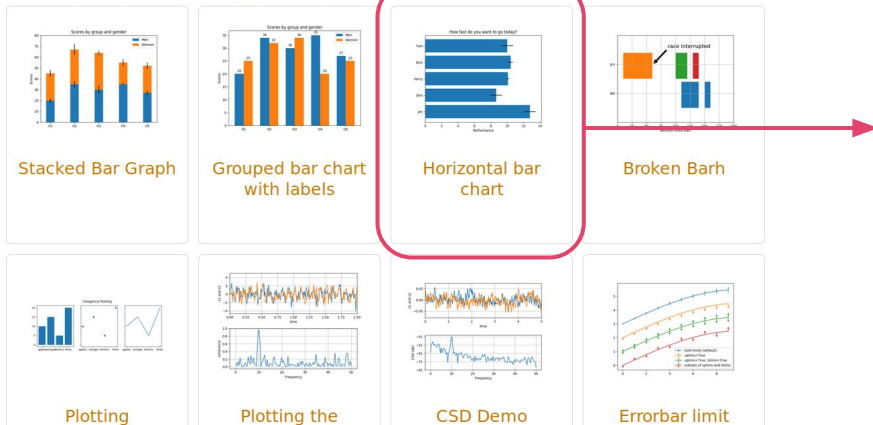
Documentación

Gallery

This gallery contains examples of the many things you can do with Matplotlib. Click on any image to see the full code.

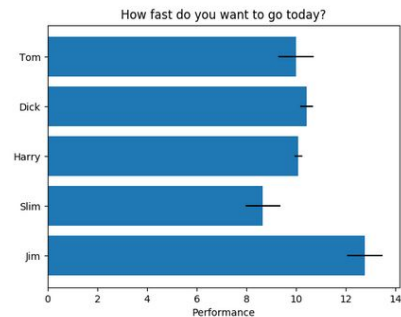
For longer tutorials, see our [tutorials page](#). You can also find [external resources](#) and a [FAQ](#) in our [user guide](#).

Lines, bars and markers



Horizontal bar chart

This example showcases a simple horizontal bar chart.



```
import matplotlib.pyplot as plt
import numpy as np

# Fixing random state for reproducibility
np.random.seed(19680801)

plt.rcParamsdefaults()
fig, ax = plt.subplots()

# Example data
people = ('Tom', 'Dick', 'Harry', 'Slim', 'Jim')
y_pos = np.arange(len(people))
performance = 3 + 10 * np.random.rand(len(people))
error = np.random.rand(len(people))

ax.barh(y_pos, performance, xerr=error, align='center')
ax.set_yticks(y_pos)
ax.set_yticklabels(people)
ax.invert_yaxis() # labels read top-to-bottom
ax.set_xlabel('Performance')
ax.set_title('How fast do you want to go today?')

plt.show()
```

Graficar en un notebook

La parte de la librería que usaremos para graficar es **matplotlib.pyplot** y se suele importar con el nombre **plt**:

```
[ ]: import matplotlib.pyplot as plt  
%matplotlib inline
```


Graficar en un notebook

La parte de la librería que usaremos para graficar es **matplotlib.pyplot** y se suele importar con el nombre **plt**:

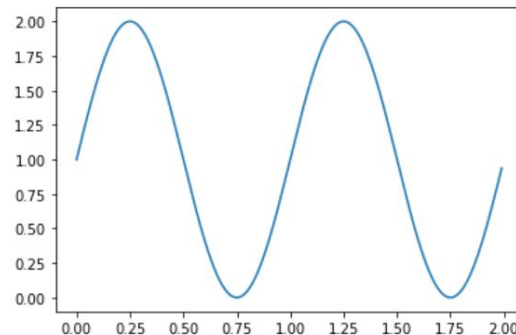
```
[ ]: import matplotlib.pyplot as plt  
  
%matplotlib inline
```

El comando mágico **%matplotlib inline** genera gráficos **incrustados** en nuestro **jupyter notebook** (últimamente ya no es necesario)



```
# En el priemr lugar va la variable horizontal y  
plt.plot(x, y)
```

[42]: [<matplotlib.lines.Line2D at 0x7f159ac8fb90>]



Crear figura y ejes

Lo primero que debemos hacer para poder graficar, es crear una **figura** y unos **ejes**.

Figura: Es el objeto correspondiente al rectángulo sobre el cual graficamos todo.

Crear figura y ejes

Lo primero que debemos hacer para poder graficar, es crear una **figura** y unos **ejes**.

Ejes: Son los distintos pares de ejes que vamos a agregar dentro de una figura.

Figura: Es el objeto correspondiente al rectángulo sobre el cual graficamos todo.

```
28]: fig = plt.figure()  
     ax = plt.axes()
```

Crear figura y ejes

Nota: Esto suena un poco abstracto al principio, pero a medida que vayamos usandolos quedará más claro que es cada uno.

Line Plot

El gráfico más sencillo que podemos hacer es **'plotear'** una **línea**.

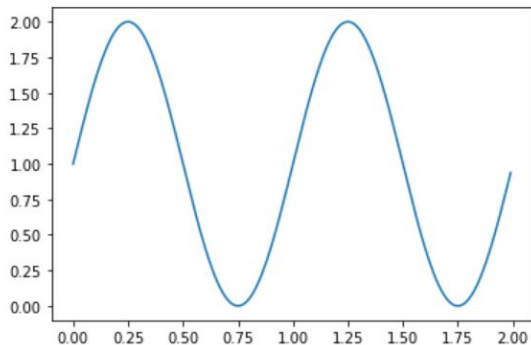
- Utilizamos función **'plot'**
- Debemos pasarle como argumento dos listas o vectores, 'x' e 'y', **del mismo tamaño**.

```
x = np.arange(0.0, 2.0, 0.01)
y = 1 + np.sin(2 * np.pi * x)
```

- El primer elemento será el que defina la coordenada horizontal y el segundo, la vertical.

```
[28]: fig = plt.figure()
      ax = plt.axes()
      ax.plot(x, y)
```

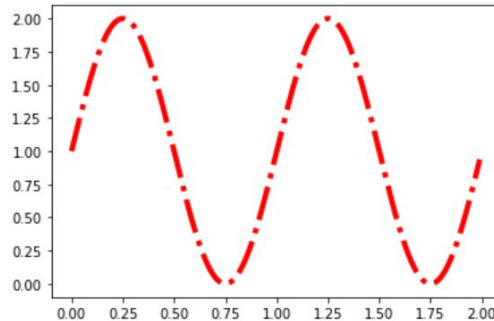
```
[28]: [<matplotlib.lines.Line2D at 0x7f159dc3bed0>]
```



Line Plot

Hay muchísimos argumentos que se le pueden pasar a la función plot para personalizar nuestro gráfico. Sólo algunos de ellos.

```
ax.plot(x, y, color = 'red', linewidth = 4, linestyle = '-.')
```



https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.plot.html

Leyendas

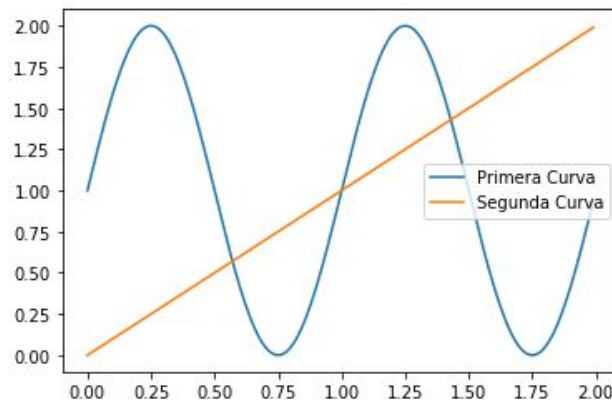


Es importante que un gráfico sea lo más **claro y explícito** posible. Para esto vamos a ayudarnos con las **leyendas**, las cuales nos permiten explicitar información sobre los distintos objetos que graficamos.

```
# Generamos la figura y los ejes
fig = plt.figure()
ax = plt.axes()

# Ploteamos las dos lineas, dandole un nombre a
ax.plot(x, y, label='Primera Curva')
ax.plot(x2,y2, label='Segunda Curva')

# Agregamos la leyenda al gráfico
ax.legend()
```



https://matplotlib.org/api/_as_gen/matplotlib.axes.Axes.legend.html#matplotlib.axes.Axes.legend

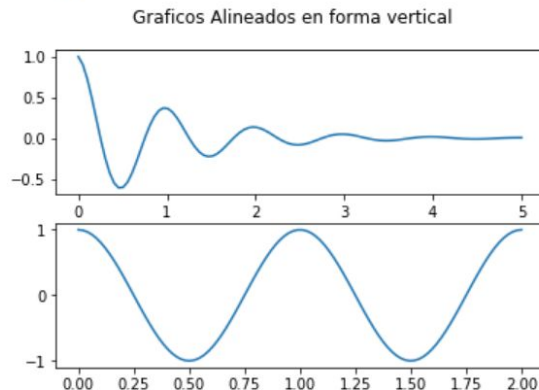
Subplots



Muchas veces vamos a querer presentar en una misma **figura** varios **axes distintos**. Esto lo podemos lograr fácilmente con la función **subplots**.

Definimos que queremos dos filas y una columna.

```
fig, axes = plt.subplots(2,1)
fig.suptitle('Graficos apilados en forma vertical')
axes[0].plot(x1, y1)
axes[1].plot(x2, y2)
```



Graficamos en cada uno de los dos ejes creados en el objeto axes. El primero en axes[0] y el segundo en axes[1].

Hands-on training



DS_Clase_06_Matplotlib.ipynb

Line Plots, Subplots y Leyendas



Continuamos con Visualización de Datos

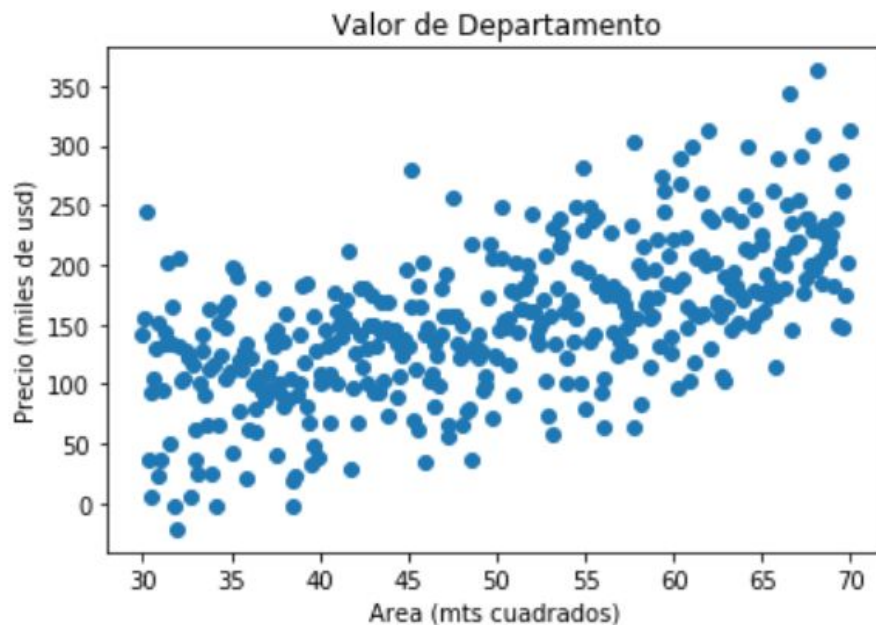


Scatter Plot

Un Scatter Plot consiste en graficar una serie de puntos, con coordenadas (x,y) , sobre un plano.

Scatter Plot

Así se ve...



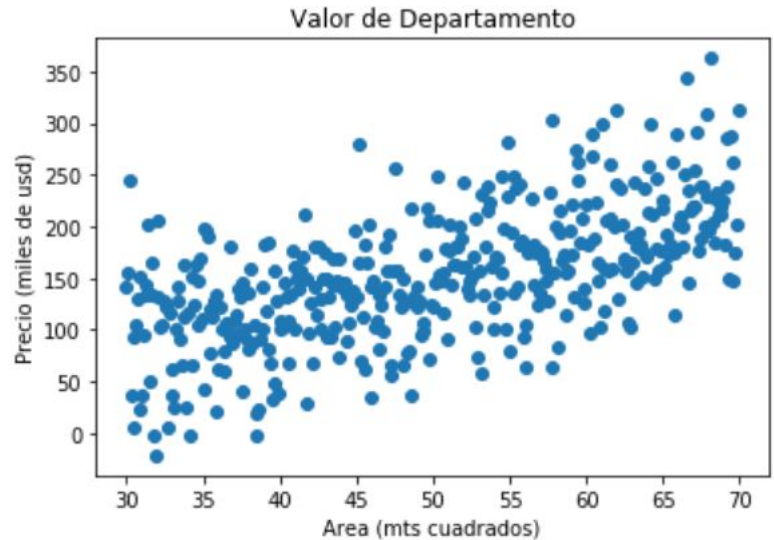
Scatter Plot

```
fig = plt.figure()
ax = plt.axes()

ax.scatter(x,y)
ax.set(xlabel='Area (mts cuadrados)',
       ylabel='Precio (miles de usd)',
       title='Valor de Departamento')
```



Podemos 'setear' los labels del eje x y el eje y, así como también el título del gráfico

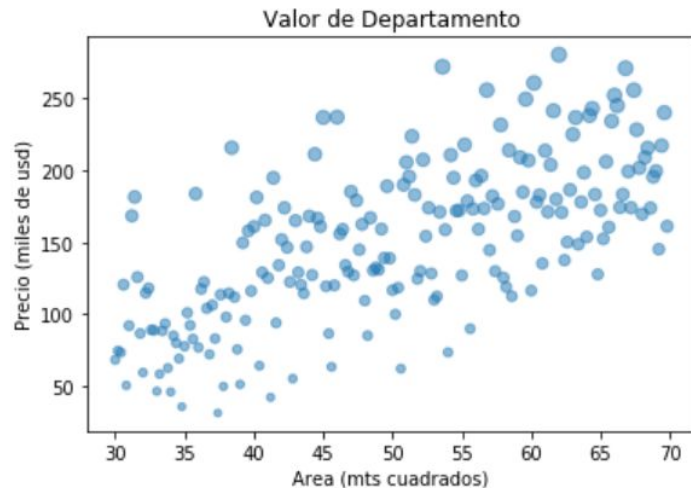


Scatter Plot

La función Scatter tiene muchísimas formas de '*customizarse*'. Les mostramos algunos ejemplos:

```
ax.scatter(x,y, s=piso, alpha=0.5)  
ax.set(xlabel='Area (mts cuadrados)',  
      ylabel='Precio (miles de usd)',  
      title='Valor de Departamento')
```

Pasamos como argumento 's' otro vector que para cada departamento (cada punto en el gráfico) tiene el número de piso. El tamaño de cada punto queda dado por el número de piso.



Scatter Plot

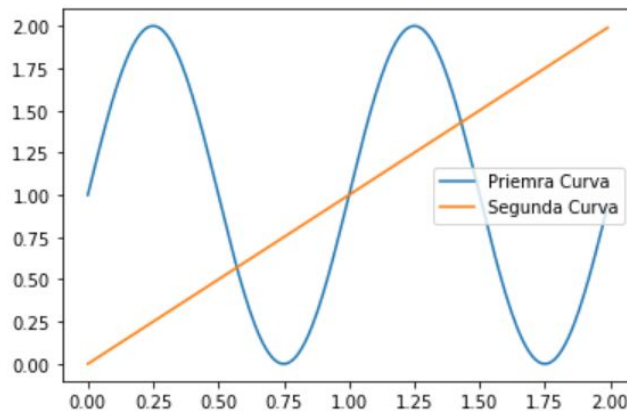
Es importante que un gráfico sea lo más **claro y explícito** posible. Para esto vamos a ayudarnos con las **leyendas**, las cuales nos permiten explicitar información sobre los distintos objetos que graficamos.



```
# Generamos la figura y los ejes
fig = plt.figure()
ax = plt.axes()

# Ploteamos las dos lineas, dandole un nombre a
ax.plot(x, y, label='Primera Curva')
ax.plot(x2,y2, label='Segunda Curva')

# Agregamos la leyenda al gráfico
ax.legend()
```



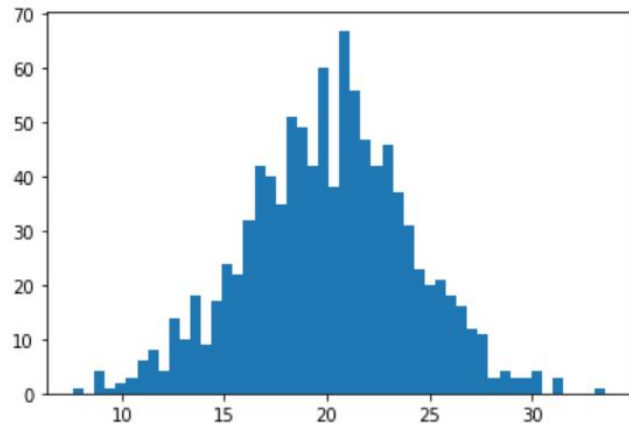
Histogramas

Matplotlib cuenta con la función **'hist'** que permite hacer **histogramas**.

Debemos pasarle como argumento una lista o vector sobre el cual queramos hacer el histograma.

Además de graficar, nos devuelve los vectores **'n'** y **'bins'**, que pueden resultarnos de utilidad para realizar cálculos.

```
# Aca decidimos la cantidad de bins que queremos  
num_bins = 50  
  
# Creamos la figura y los ejes  
fig, ax = plt.subplots()  
  
# Ploteamos el histograma  
n, bins, _ = ax.hist(x, num_bins)
```



Hands-on training



DS_Clase_06_Matplotlib.ipynb

Scatter Plots e Histogramas



Para la próxima

1. Ver los videos de la plataforma **“Visualización de datos”**, en particular los de **Seaborn**.
2. Completar los **notebooks** de hoy (¡y atrasados!)
3. Mirar la consigna de la **Entrega 01**
4. Tal vez vengan un poco abrumados/as. La semana que viene vamos a aprovechar a repasar y a hacer ejercicios integradores. Salvo por Seaborn, no daremos contenidos nuevos.

ACÀMICA