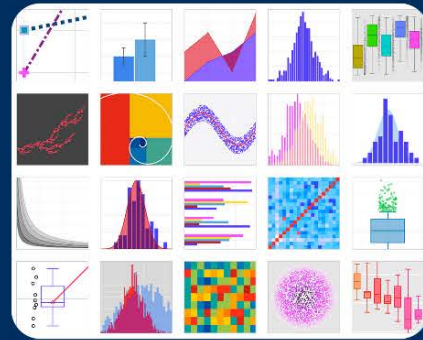


Introducción al Análisis y Visualización de Datos en Python

Día 4 – Matemáticas Simbólicas y Visualización de Datos



Presentan:

Dr. Ulises Olivares Pinto

Walter André Rosales Reyes

Escuela Nacional de Estudios Superiores Unidad Juriquilla



Contenido

1. Visualización de datos y Matemáticas Simbólicas

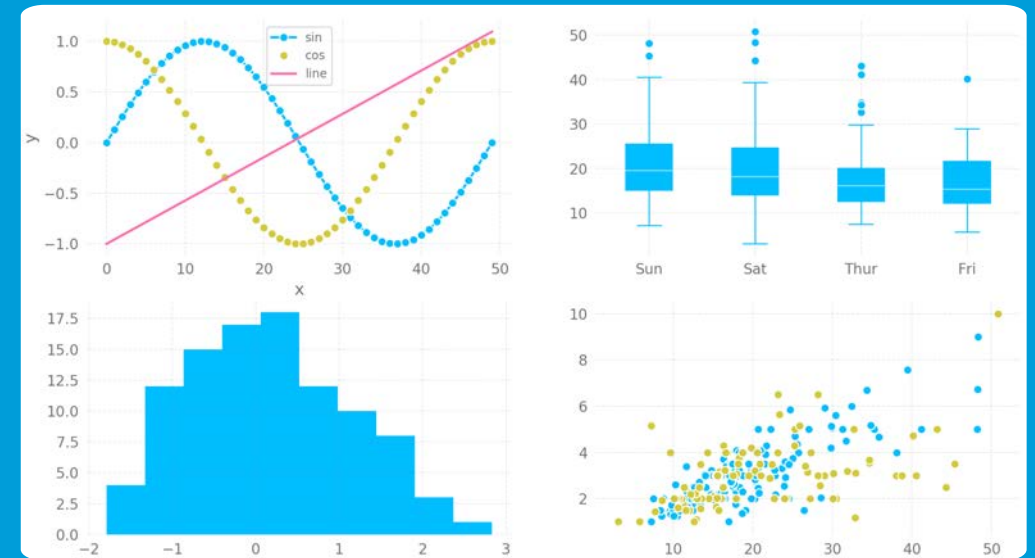
(~3 horas - 2 bloques)

- ¿Cómo elegir un gráfico adecuado?
- Matplotlib
- Manipulaciones algebraicas
- Cálculo
 - Expansión de series
 - Diferenciación
 - Integración
 - Resolución de ecuaciones algebraicas

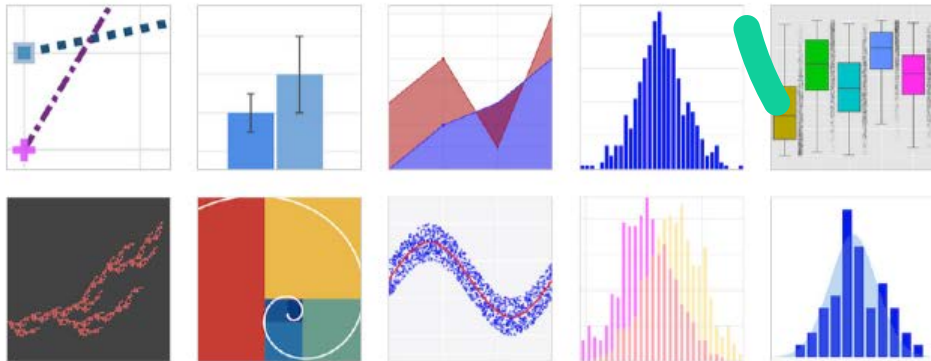
2. Proyecto

(30 ~45 minutos)

Visualización de datos con

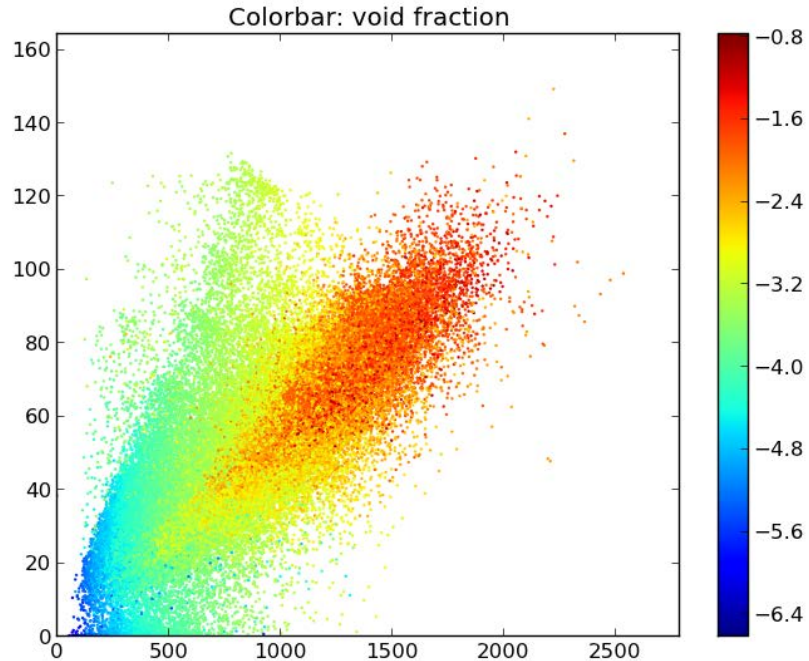


Tipos de gráficos



El tipo de gráfico es un aspecto muy importante, debido a que es el mensaje que se proyectará. Entonces, se deber seleccionar el gráfico de acuerdo a los datos con los que se cuenta .

Gráficos de dispersión (scatter)



- Mostrar variables cuantitativas en un intervalo específico.
- Los puntos se deberán distinguir claramente.
- Si existe más de una categoría de deberá utilizar colores y formas para hacer la distinción entre ambas categorías.

Diagrama Dispersión

Utilizar dataframe `salary.csv` para generar un gráfico de dispersión.

```
# importar la librería de pandas como pd
import pandas as pd

#Leer el dataframe salaries.csv
df = pd.read_csv("https://raw.githubusercontent.com/ulises1229/INTRO-PYTHON-ENESJ/master/data/salaries.csv")
df.head()
```


Diagrama Dispersión

```
# Almacenar variables años y salario
x = df.service
y = df.salary

# Importar librería Matplotlib
import matplotlib.pyplot as plt

# Generar diagrama de dispersión
plt.scatter(x, y)

# Títulos: gráfico y ejes
plt.title("Gráfico de Dispersión Años de servicio vs Salario")
plt.xlabel('Años de servicio')
plt.ylabel('Salario Anual')

# Desplegar imagen en pantalla
plt.show()
```

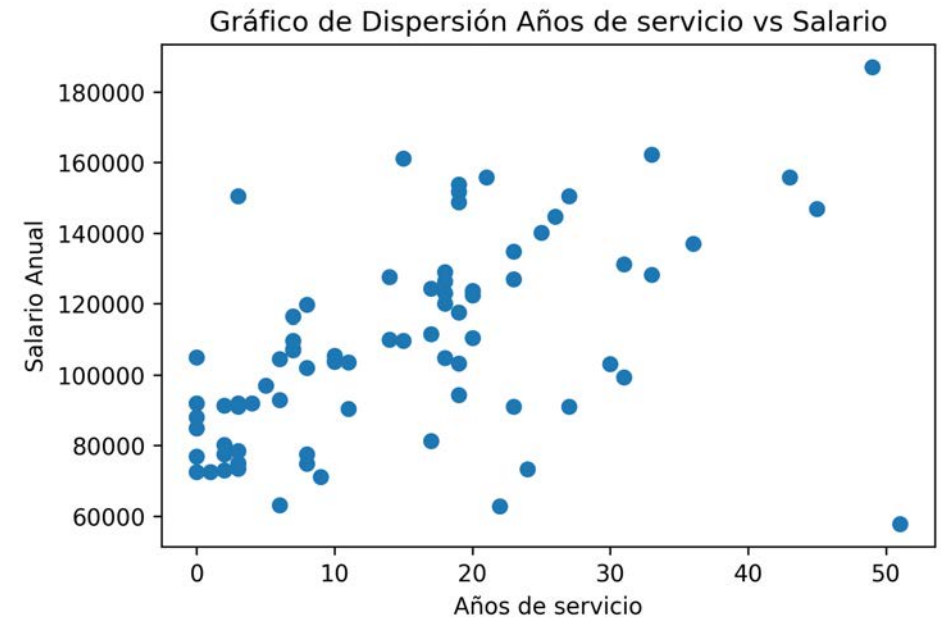


Diagrama Dispersión

- Agregar colores y tamaño.

```
# Importar librería de numpy
import numpy as np

# Almacenar variables x = años servicio y y = salario
x = df.service
y = df.salary

# Arreglo de colores
colores = np.arange(len(x))
print(colores)

# Agregar un scatter con colores y transparencia
plt.scatter(x, y, c=colores, alpha = 0.5, s = (y*0.002))
```

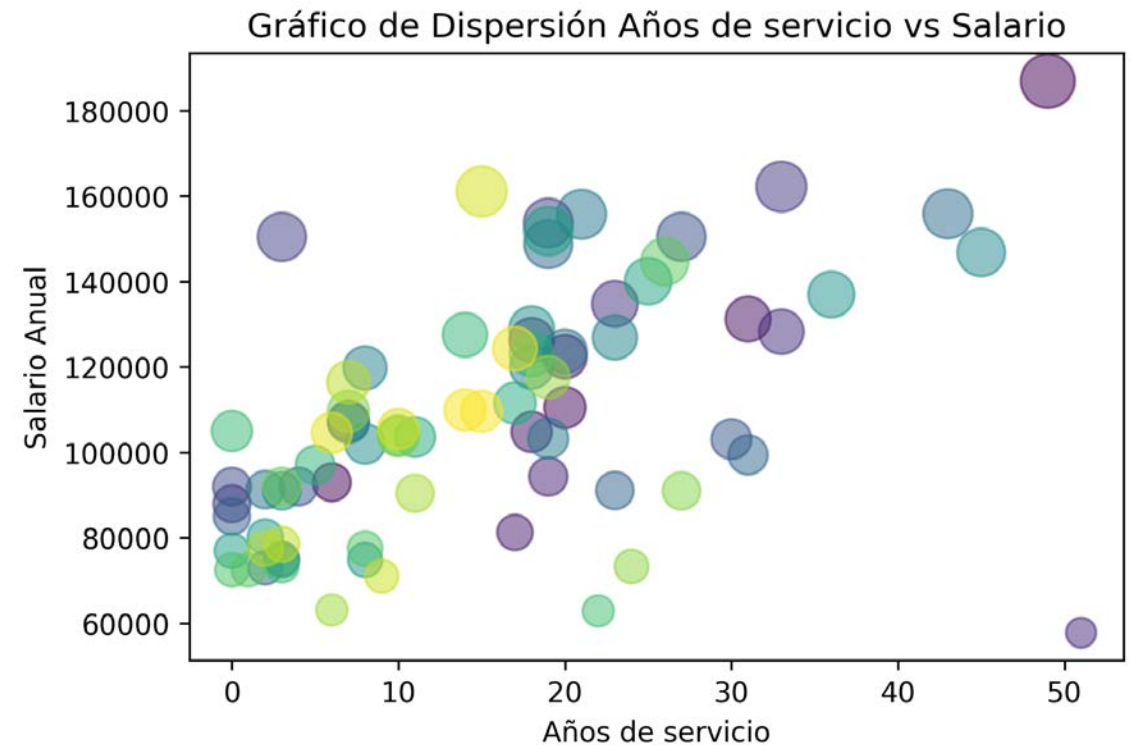


Diagrama Dispersión

- Agregar colores, tamaño y leyenda.

```
# Almacenar variables x = años servicio y y = salario
x = df.service
y = df.salary
lab = ['Male', 'Female']

# Arreglo de colores
colores = np.arange(len(x))

# Títulos: gráfico y ejes
plt.title("Gráfico de Dispersión Años de servicio vs Salario")
plt.xlabel('Años de servicio')
plt.ylabel('Salario Anual')

# Agregar un scatter con colores, transparencia y leyenda
plt.scatter(x, y, c=colores, alpha = 0.5, s = (y*0.002), label = "datos")
plt.legend(loc='best')

plt.show()
```

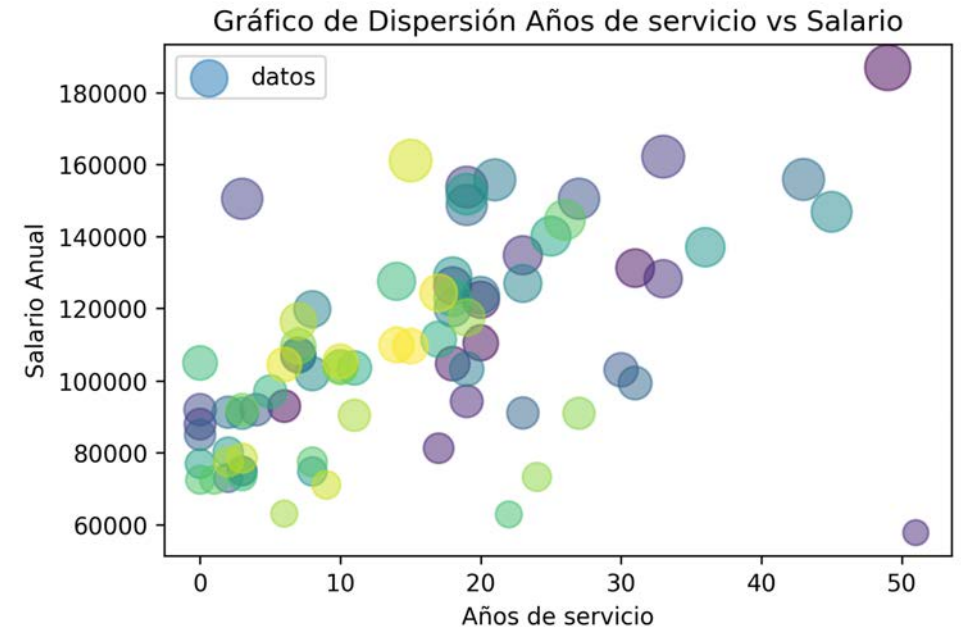
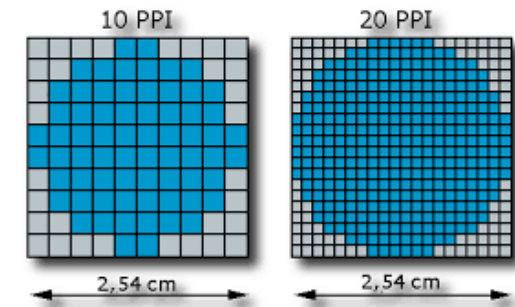
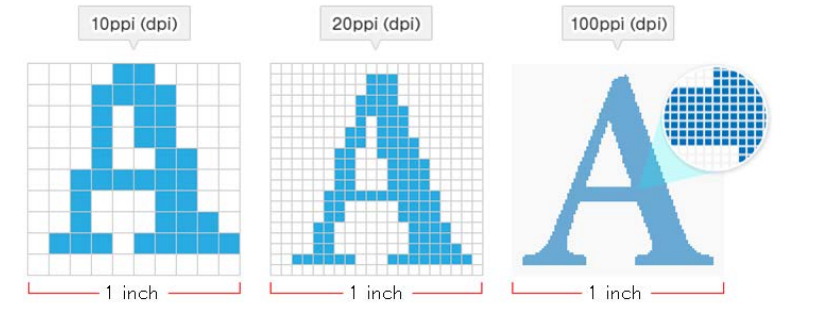


Diagrama de Dispersión – Exportando un Gráfico

Con matplotlib es posible exportar imágenes de alta calidad. Para esto es necesario definir elementos tales como:

- Título de gráfico
- Títulos de los ejes
- Título de la leyenda

```
# guarda la imagen en un archivo  
plt.savefig("dispersión.tiff", dpi=300, quality = 95, bbox_inches='tight')
```



Ejercicio 1

Emplear el dataframe **mpg.csv** Está en el repositorio de GitHub

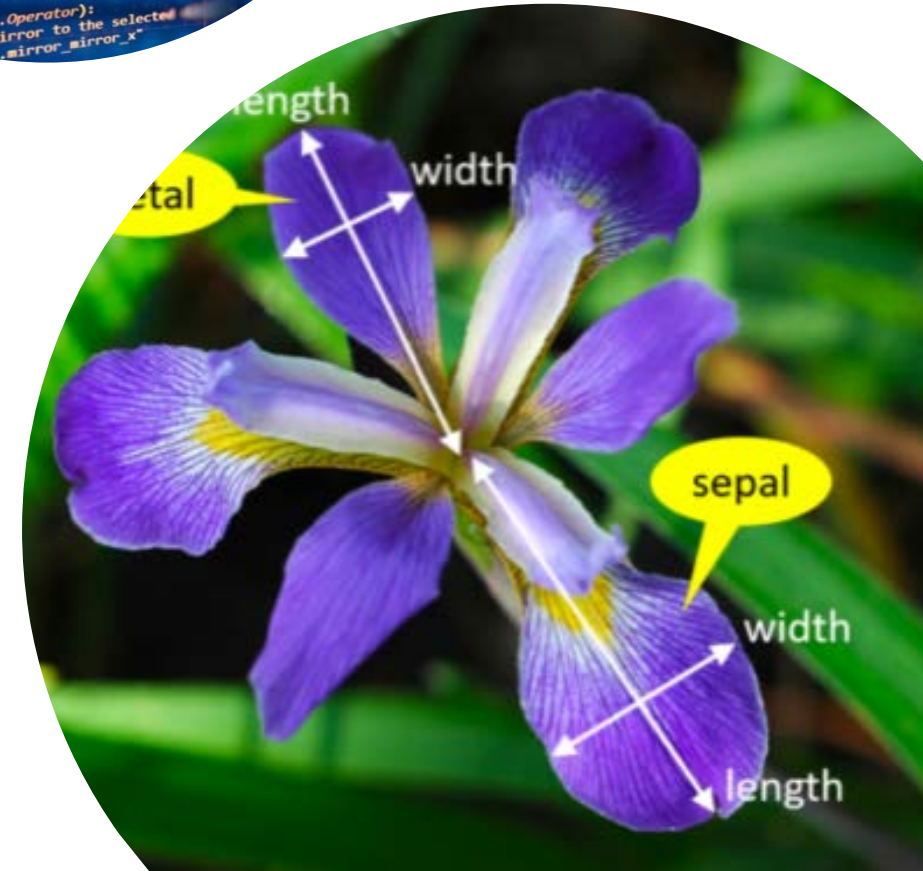
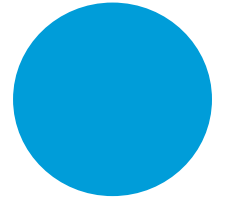
1. Se deberá generar un gráfico de dispersión del desplazamiento del motor vs rendimiento en autopista.
2. Se deberá generar un segundo gráfico de desplazamiento del motor vs rendimiento en ciudad.

Ejercicio 2

Emplear el dataframe **iris.csv**

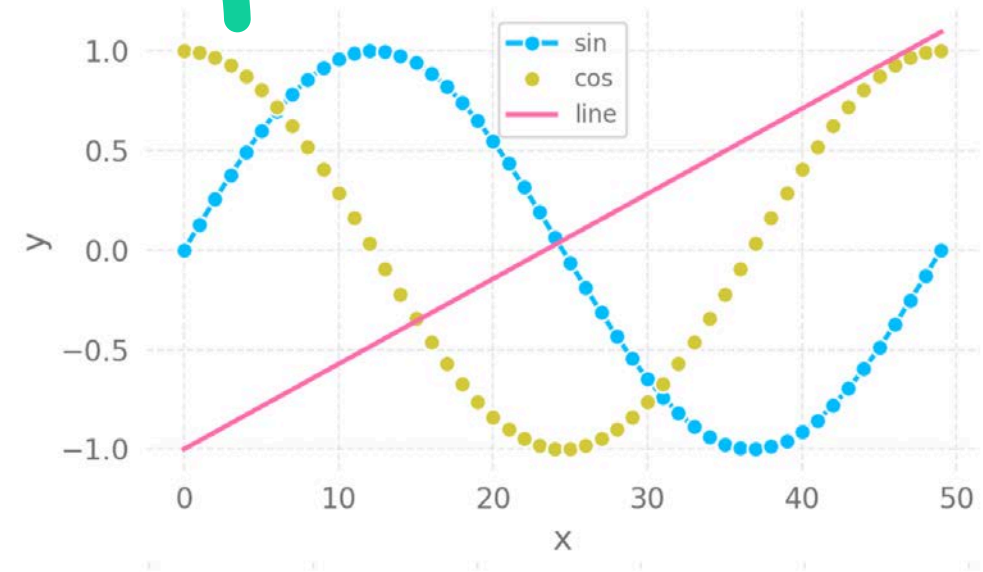
Está en el repositorio de GitHub

1. Se deberá generar un gráfico de dispersión de **Longitud de pétalo** vs **Longitud de sepalo**. Lo anterior se deberá hacer para cada especie, se deberá poner el nombre de la especie en una leyenda.



Gráficos de Línea

- Representar valores cuantitativos en función de la variable independiente.
- Es posible combinar la gráfica con puntos, para enfatizar valores específicos.
- Si existe más de una categoría en la gráfica, deberán utilizarse leyendas y tipos de línea para diferenciar los datos.



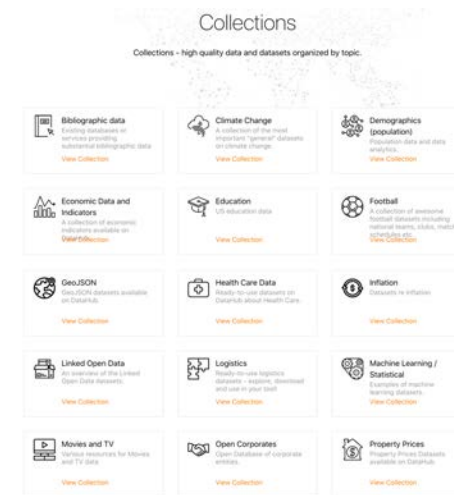
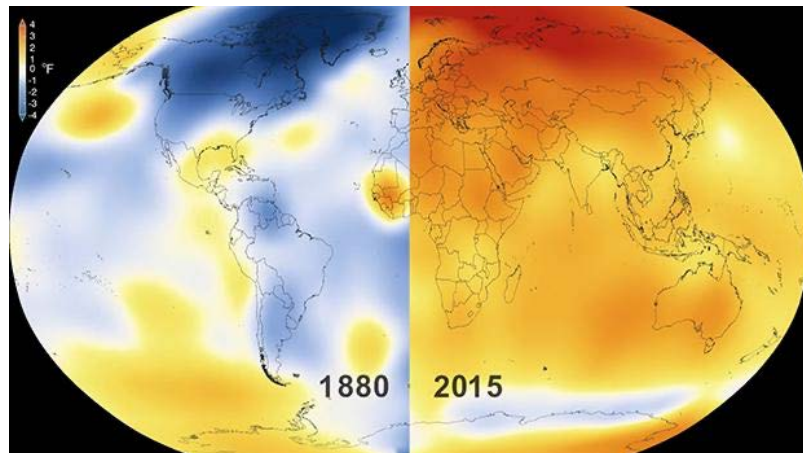
Gráficos de línea

- Se deberá importar el dataframe temp.csv (Repositorio GH)

```
# Importar pandas y matplotlib
import matplotlib.pyplot as plt
import pandas as pd

# Importar data frame de temperatura
df_temp = pd.read_csv('https://raw.githubusercontent.com/ulises1229/INTRO-PYTHON-ENESJ/master/data/temp.csv', sep=',')

df_temp
```



Gráficos de línea

```
# Importar pandas y matplotlib
import matplotlib.pyplot as plt
import pandas as pd

# Importar data frame de temperatura
df_temp = pd.read_csv('https://raw.githubusercontent.com/ulises1229/INTRO-PYTHON-ENESJ/master/data/temp.csv', sep=',')

df_temp

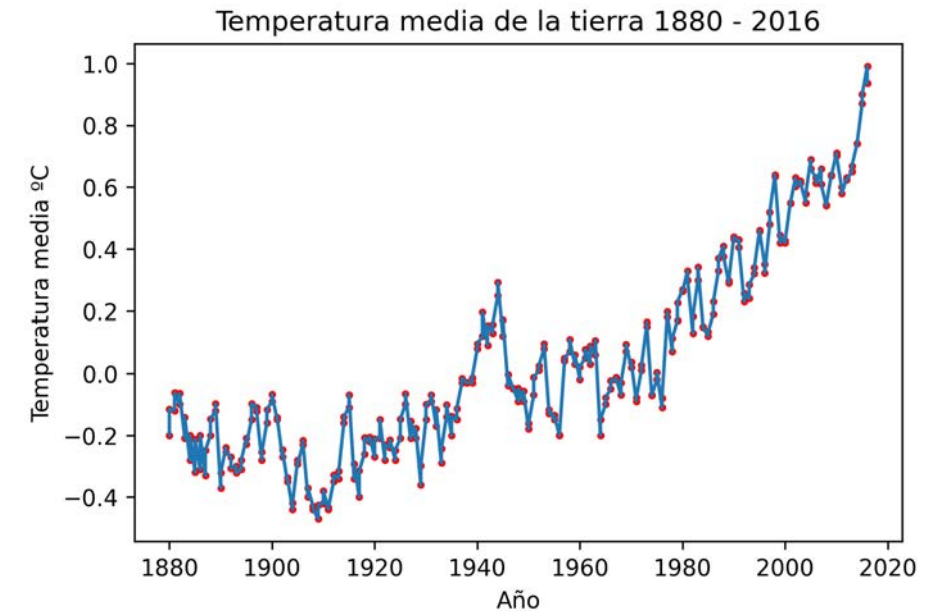
# Separa dos variables del dataframe
año = df_temp.Year
temp = df_temp.Mean

# Generar el diagrama de línea
plt.plot(año, temp)

# Agregar un scatter
plt.scatter(año, temp, s=5, c="red")

# Agregar títulos
plt.title("Temperatura media de la tierra 1880 - 2016")
plt.xlabel('Año')
plt.ylabel('Temperatura media °C')

# Guardar el gráfico en un archivo
plt.savefig("line_temp.tiff", dpi=300, quality=95, bbox_inches='tight')
```



Ejercicio 3

Emplear el dataframe **co2.csv**

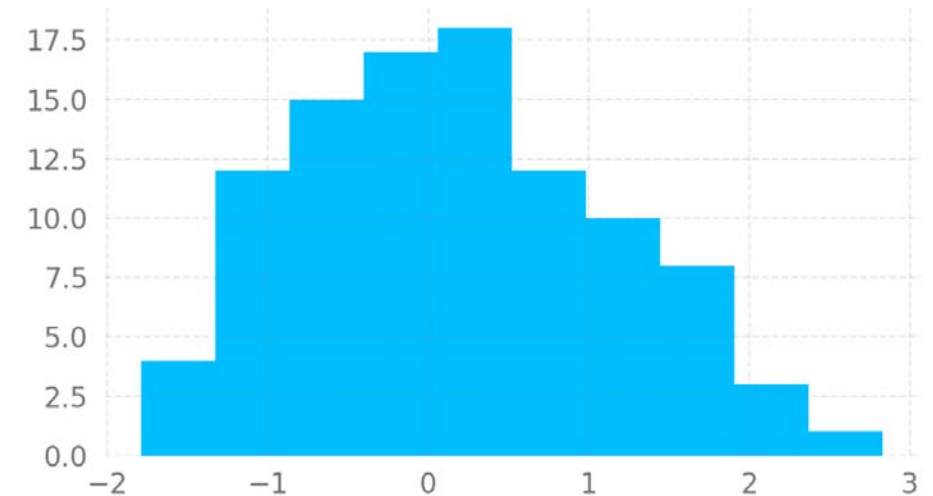
Está en el repositorio de GitHub

1. Se deberá generar un gráfico de línea con **fecha** vs **Interpolated**.
2. Se deberá generar un gráfico de línea con **fecha** vs **Trend**.
3. Se deberá exportar el gráfico con una calidad de 300 dpi.



Gráficos de barras

- Mostrar comparaciones, desviaciones o rankings. Puede estar en formato vertical u horizontal.
- Utilizar solo un color, a menos que haya más de una categoría. En dado caso se usan leyendas.
- Evitar utilizar bordes para las barras.
- Distancia entre barras, deberá ser suficiente para que no se traslapen.



Gráficos de barras

```
# Importar pandas y matplotlib
import matplotlib.pyplot as plt
import pandas as pd

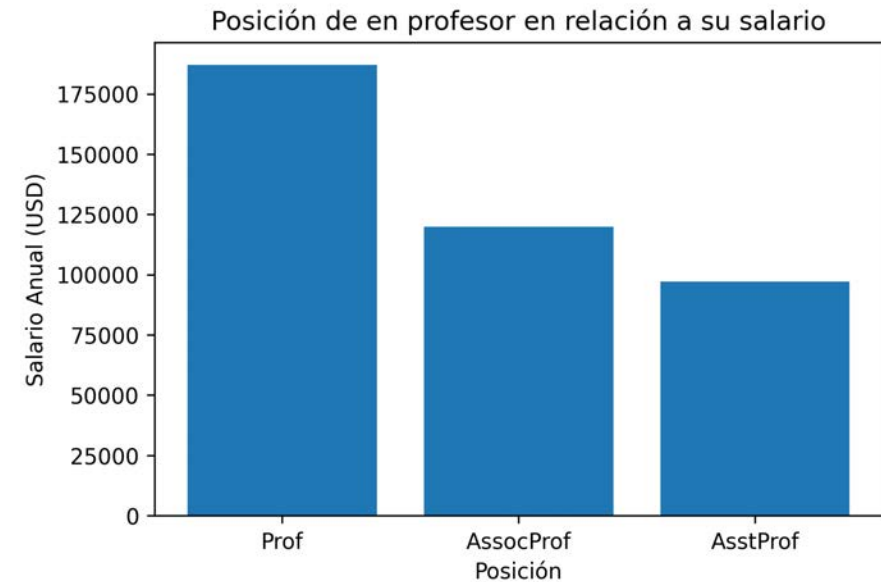
#Leer el dataframe salaries.csv
df = pd.read_csv("https://raw.githubusercontent.com/ulises1229/INTRO-PYTHON-ENESJ/master/data/salaries.csv")

# Recuperar rank y salario
rank = df['rank']
sueldo = df.salary

# Generar una gráfica de barras
plt.bar(rank, sueldo)

# Agregar títulos
plt.title("Posición de en profesor en relación a su salario")
plt.xlabel("Posición")
plt.ylabel("Salario Anual (USD)")

plt.savefig("salario.tiff", dpi=300, quality = 95, bbox_inches='tight')
```



Ejercicio 4

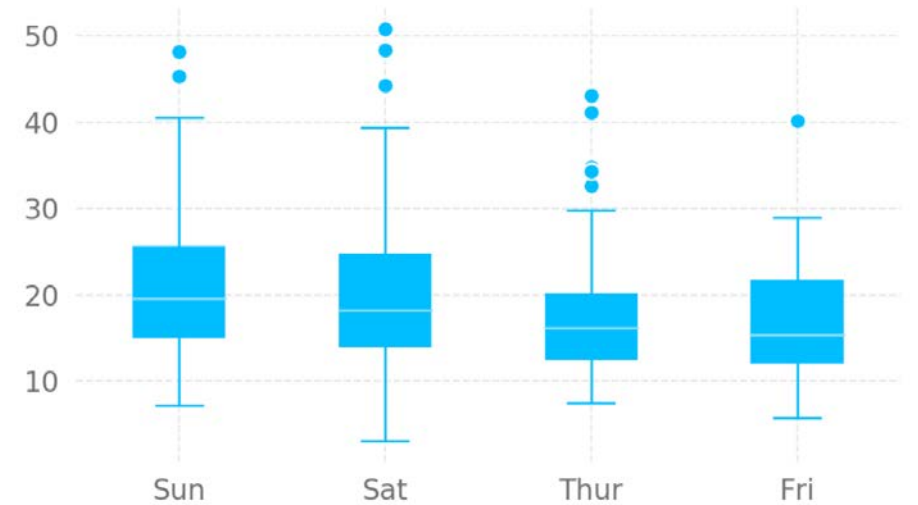
Emplear el dataframe **salaries.csv**

Está en el repositorio de GitHub

1. Se deberá generar un gráfico de barras comparando el salario por sexo.

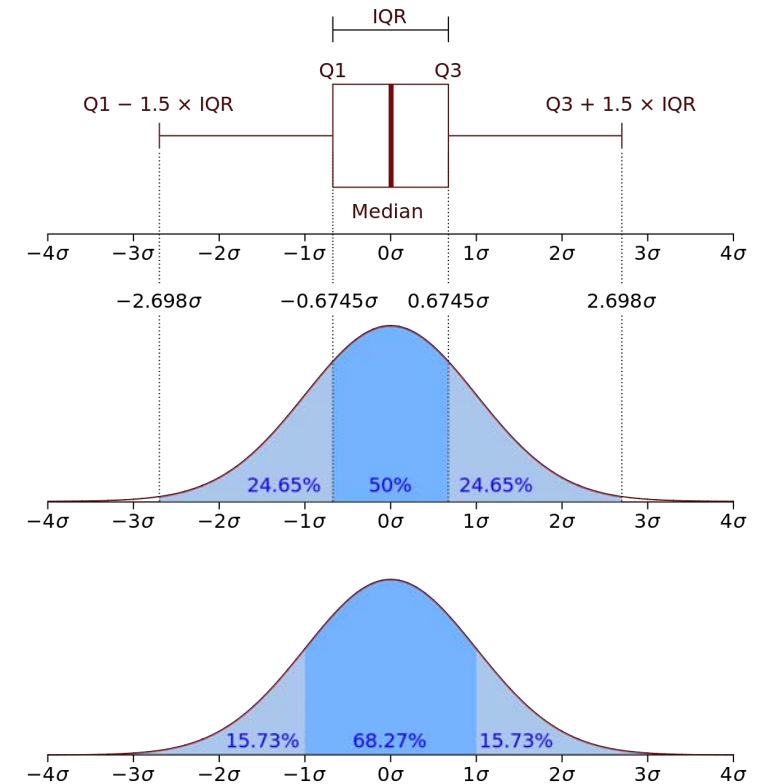
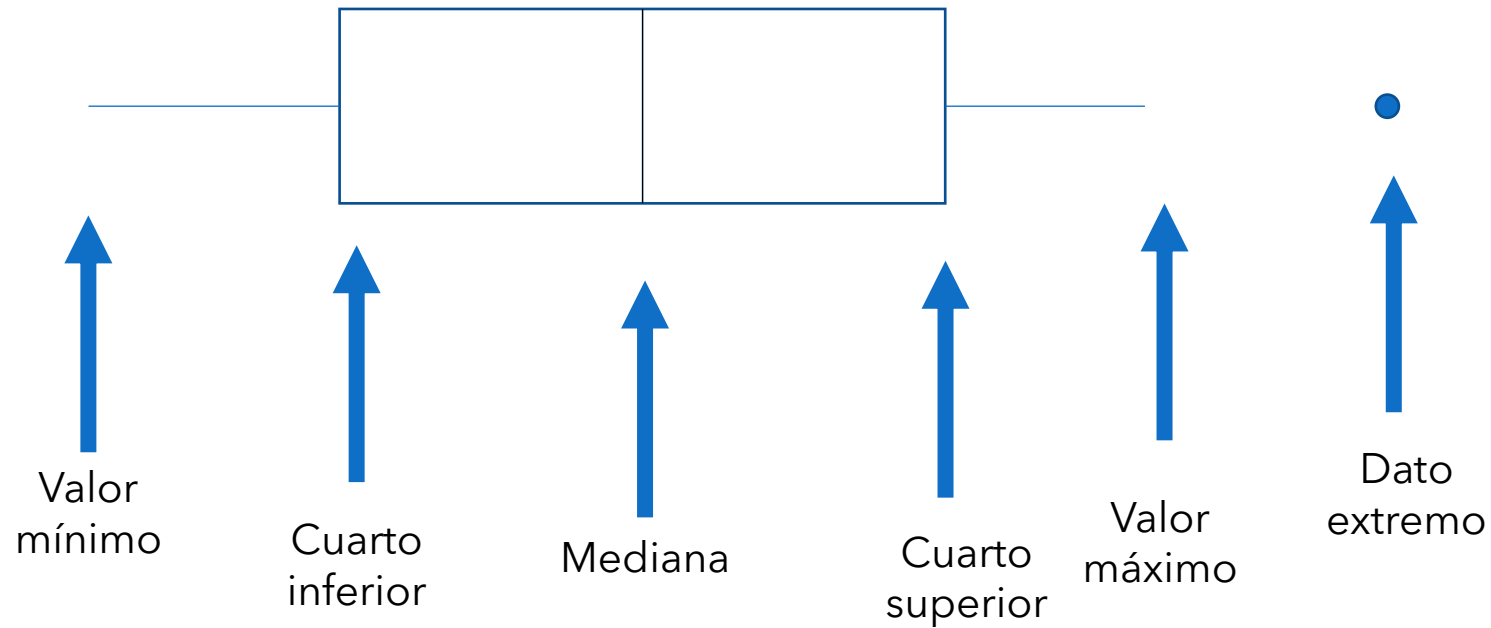
Boxplots

- y variable numérica
- x variable nominal



Boxplots

Cuartiles: Valores que dividen la muestra en 4 partes iguales



Boxplots

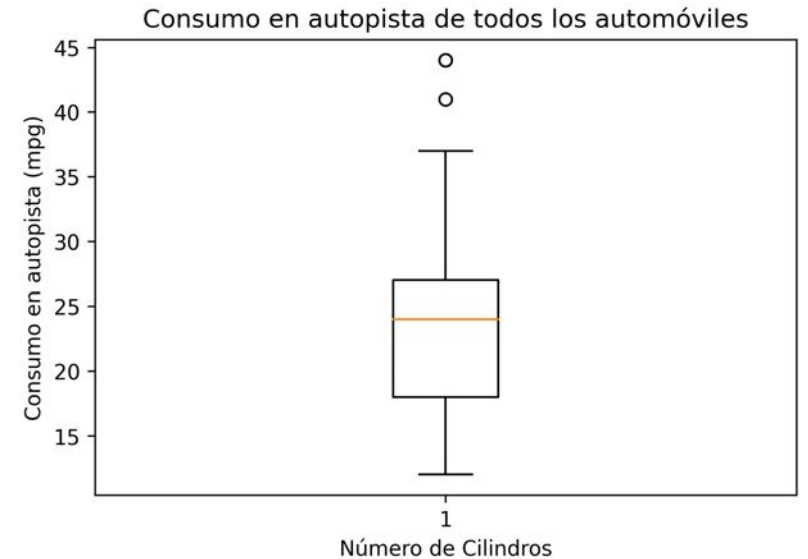
```
# Importar pandas y matplotlib
import matplotlib.pyplot as plt
import pandas as pd

# importar ma

#Leer el dataframe salaries.csv
mpg = pd.read_csv("https://raw.githubusercontent.com/ulises1229/INTRO-PYTHON-ENESJ/master/data/mpg.csv")

plt.boxplot(mpg.hwy)
# Agregar títulos
plt.title("Consumo en autopista de todos los automóviles")
plt.xlabel("Número de Cilindros")
plt.ylabel("Consumo en autopista (mpg)")

# exportar imagen
plt.savefig("mpg.tiff", dpi=300, quality = 95, bbox_inches='tight')
```



Ejercicio 5

Emplear el dataframe [mpg.csv](#)

Está en el repositorio de GitHub

1. Se deberá generar un bloxplot para el rendimiento en ciudad.
2. Se deberá generar un bloxplot para el desplazamiento.

Matemáticas Simbólicas en Python



SymPy



Una Variable Simbólica

```
# Importar la librería Sympy
from sympy import *

# importar una variable simbólica (x)
from sympy.abc import x

# Habilitar impresión avanzada
init_printing(use_latex='mathjax')

# Lo anterior es equivalente a:
x = Symbol('x')

# también se puede definir como:
x, y = symbols('x y')
```

Operaciones Básicas

```
[178] # Definir una expresión simple
```

```
a = 2*x**2 + y
```

```
display(a)
```

↳

$$2x^2 + y$$

```
[168] # Operaciones simples con expresiones
```

```
display(a+y)
```

```
display(a**2 + y**2)
```

↳

$$y^2 + (2x^2 + y)^2$$

```
[161] # Expandir una expresión
```

```
expand(a**3)
```

↳

$$8x^6 + 12x^4y + 6x^2y^2 + y^3$$

```
[171] # Factorización
```

```
factor(x**3 + 3*x**2 + 3*x + 1)
```

↳

$$(x + 1)^3$$

Operaciones Básicas

```
[177] # Simplificación de expresiones
      # Definir dos expresiones
      exp1 = x**2 - y**2
      exp2 = (x+y)*(x-y)

      #
      display("Primer expresión: ",exp1)
      display("Segunda expresión: ",exp2)
      display("Simplificación: ", simplify(exp1 - exp2))
```

☞ 'Primer expresión: '

$$x^2 - y^2$$

'Segunda expresión: '

$$(x - y)(x + y)$$

'Simplificación: '

$$0$$

Sustituciones

```
[182] # Definir una expresión simple y realizar sustituciones
expr = cos(x) + 1
display("Primer sustitución x = y", expr.subs(x, y))
display("Segunda sustitución x = 0", expr.subs(x, 0))
```

↳ 'Primer sustitución'

$\cos(y) + 1$

'Segunda sustitución'

2

```
# Continuación sustituciones
expr = x**y
display("Expresión original", expr)
expr = expr.subs(y, x**y)
display("Después de la sustitución", expr)
```

↳ 'Expresión original'

x^y

'Después de la sustitución'

x^{x^y}

Funciones e identidades Trigonométricas

```
[191] # Definición de una expresión trig.  
      trigx = sin(2*x) + cos(2*x)  
  
      # Expansión utilizando identidades  
      display("Expansión ", expand_trig(trigx))
```

☞ 'Expansión '

$$2 \sin(x) \cos(x) + 2 \cos^2(x) - 1$$

```
[193] # Simplificación  
      display("Simplificación ", trigsimp(cos(x)**2 + sin(x)**2))
```

☞ 'Simplificación '

Simplificación y Evaluación

```
[6] # Definición de una expresión
    expr = sqrt(8)

    # imprimir la expresión
    display(expr)

    # Imprimir la evaluación
    display(expr.evalf())

    # imprimir pi y hacer su evaluación
    display("Símbolo ", pi)
    display("Evaluación ", pi.evalf(100))
```



$$2\sqrt{2}$$

2.82842712474619

'Símbolo '

π

'Evaluación '

3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628034825342117068

Simplificación



Más simplificaciones

```
display("",simplify((x**3 + x**2 -x -1)/(x**2 +2*x +1)))  
display("",simplify((x**4 - 1)/(x - 1)))  
display("",simplify((x**4 - 1)/(x**2 - 1)))  
display("",expand((x+2*y)**3))  
display("",cancel((x**4 - 1)/(x - 1)))  
display("",cancel((x**2 + 2* x + 1)/(x**2 - 1)))
```



''

$$x - 1$$

''

$$\frac{x^4-1}{x-1}$$

''

$$x^2 + 1$$

''

$$x^3 + 6x^2y + 12xy^2 + 8y^3$$

''

$$x^3 + x^2 + x + 1$$

''

$$\frac{x+1}{x-1}$$

Exponenciales y logaritmos

```
# Definición de un nuevo símbolo
n = Symbol('n')

# Logaritmos
display('', ln(x))
display('', expand_log(log(x*y)))
display('', expand_log(log(x/y)))
display('', expand_log(log(x**2)))
display('', expand_log(log(x**n)))
```

↳ ''	$\log(x)$
''	$\log(xy)$
''	$\log\left(\frac{x}{y}\right)$
''	$\log(x^2)$
''	$\log(x^n)$

Cálculo – Derivación

```
#Diferenciación
# definiendo una función
f = x**3 + 2*x**2 + 3*x + 5
display("Función: ", f)

# Calculando primer derivada
derivada1 = diff(f, x)
display("1er Derivada: ", derivada1)

# Calculando segunda derivada
derivada2 = diff(derivada1, x)
display("2da Derivada: ", derivada2)
```

☞ 'Función: '

$$x^3 + 2x^2 + 3x + 5$$

'1er Derivada: '

$$3x^2 + 4x + 3$$

'2da Derivada: '

$$6x + 4$$

Cálculo – Integración

```
[52] #Integración
      # definiendo una función
      f = x**3 + 2*x**2 + 3*x + 5
      display("Función: ",f)

      # Intervalos de evaluación
      a = 1
      b = 5

      # Imprimir la integral
      display("",Integral(f,(x,a,b)))

      # Imprimir la evaluación
      display("",integrate(f,(x,a,b)))
```

↪ ''
''

$$\int_1^5 (x^3 + 2x^2 + 3x + 5) dx$$

$$\frac{884}{3}$$

Cálculo – Límites

```
[54] # Límites
      # definiendo una función
      f = x**3 + 2*x**2 + 3*x + 5
      display("Función: ", f)

      # Evaluación del límite cuando x -> inf
      display("Límite", limit(f, x, oo))
```

↳ 'Función: '

'Límite'

$$x^3 + 2x^2 + 3x + 5$$

27

Ecuaciones y Raices



```
# Ecuaciones
# definiendo una ecuación  $x^3 + 2x^2 + 3x + 5 = 0$ 
f = x + 2

display("Resolver ecuación ", solve(f, x))

display("Raices de un polinomio", roots(x**3 - 6*x**2 + 9*x, x))
```

↳ 'Resolver ecuación '

$[-2]$

'Raices de un polinomio'

$\{0:1, 3:2\}$