

FACULTAD DE
INGENIERÍA Y CIENCIAS



UAI
UNIVERSIDAD ADOLFO IBÁÑEZ

INTRODUCCIÓN A PYTHON

PANDAS Y GRÁFICOS

Miguel Carrasco
miguel.carrasco@uai.cl

- ▶ Introducción a Pandas
 - Operaciones básicas de datos
 - Agrupaciones de datos
 - Pivot table
 - CrossTab
 - Gráfica matplotlib
 - Gráfica Plotly



```
self.FidValue = OrderedDict(sorted(self.items(), key=lambda item: item[0]))
#Read item in dictionary
for key, value in item.FidValue.items():
    typeOfFID = mapFidType[key]
    if(typeOfFID == "DATE"):
        d = datetime.datetime.strptime(str(value), "%Y-%m-%d")
        dataCal = datetime.date.strptime(str(value), "%Y-%m-%d")
        FidAndValue = FidAndValue + value
    else:FidAndValue = FidAndValue + value
```

```
try:
    start = date(int(self.start_year.get(self.months.index(self.start_month)),
                int(self.start_day.get(self.months.index(self.start_month))),
                int(self.start_year.get(self.months.index(self.start_month))))
    end = date(int(self.end_year.get(self.months.index(self.end_month)),
                int(self.end_day.get(self.months.index(self.end_month))),
                int(self.end_year.get(self.months.index(self.end_month))))
```

```
import pandas as pd
import matplotlib.pyplot as plt

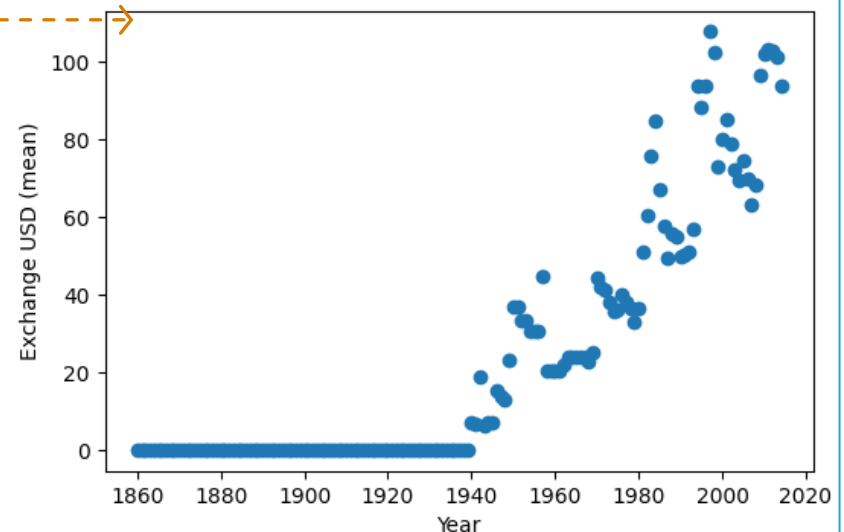
df = pd.read_csv('african_crises.csv')
print(df.columns)

group_1 = df.groupby(['year'])
stats_1 = group_1['exch_usd'].agg(['mean', 'std'])

x = stats_1.index
y = stats_1['mean']

plt.scatter(x,y)
plt.xlabel('Year')
plt.ylabel('Exchange USD (mean)')
plt.show()
```

scatter() nos permite graficar puntos, en este caso emplearemos index, y la media de los pesos



```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('african_crises.csv')
print(df.columns)

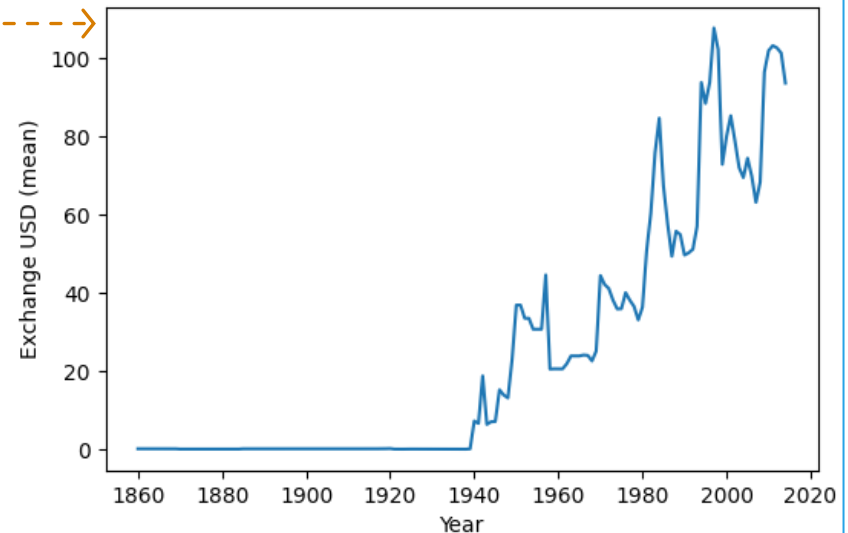
group_1 = df.groupby(['year'])
stats_1 = group_1['exch_usd'].agg(['mean', 'std'])
```

```
x = stats_1.index
y = stats_1['mean']
```

```
plt.plot(x,y)
```

```
plt.xlabel('Year')
plt.ylabel('Exchange USD (mean)')
plt.show()
```

plot() nos permite graficar líneas entre los puntos x,y



```
import pandas as pd
import matplotlib.pyplot as plt

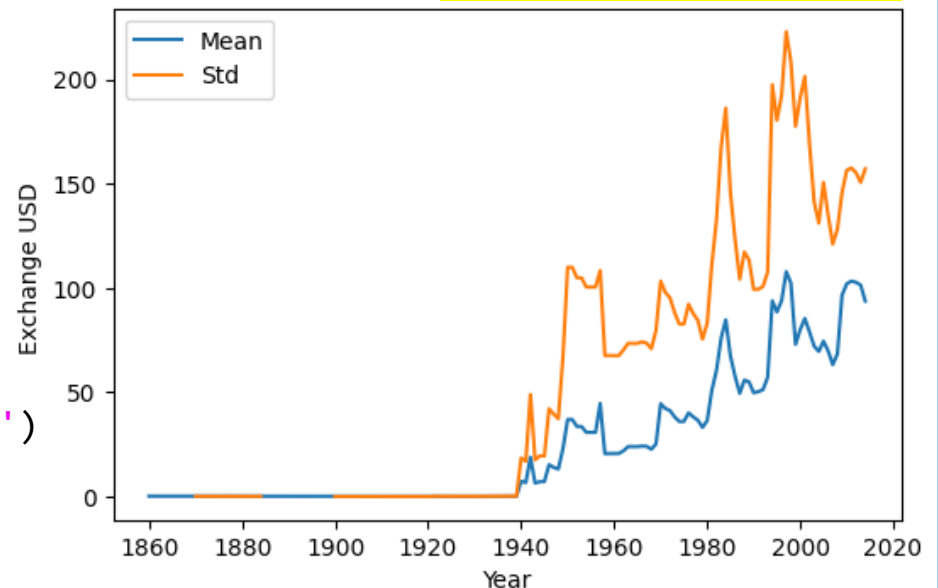
df = pd.read_csv('african_crises.csv')
print(df.columns)
group_1 = df.groupby(['year'])
stats_1 = group_1['exch_usd'].agg(['mean', 'std'])
```

```
x = stats_1.index
yA = stats_1['mean']
yB = stats_1['std']

plt.plot(x,yA,label='Mean')
plt.plot(x,yB,label='Std')

plt.xlabel('Year')
plt.ylabel('Exchange USD (mean)')
plt.legend()
plt.show()
```

Podemos graficar sobre el mismo gráfico con múltiples líneas



```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('african_crises.csv')
print(df.columns)
group_1 = df.groupby(['year'])
stats_1 = group_1['exch_usd'].agg(['mean', 'std'])

x = stats_1.index
yA = stats_1['mean']
yB = stats_1['std']
plt.plot(x,yA,label='Mean')
plt.plot(x,yB,label='Std')

plt.xlabel('Year')
plt.ylabel('Exchange USD (mean)')
plt.legend()
plt.savefig('figura.png')

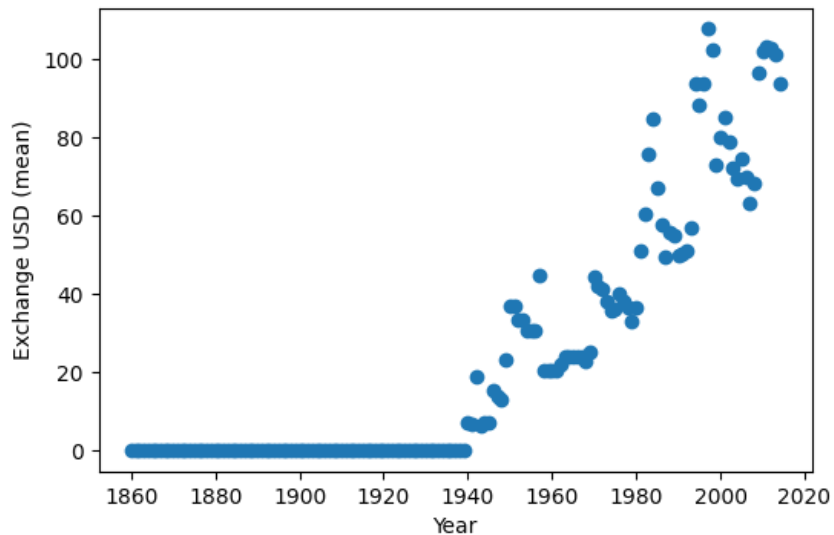
plt.show()
```

savefig() permite guardar el gráfico en un archivo



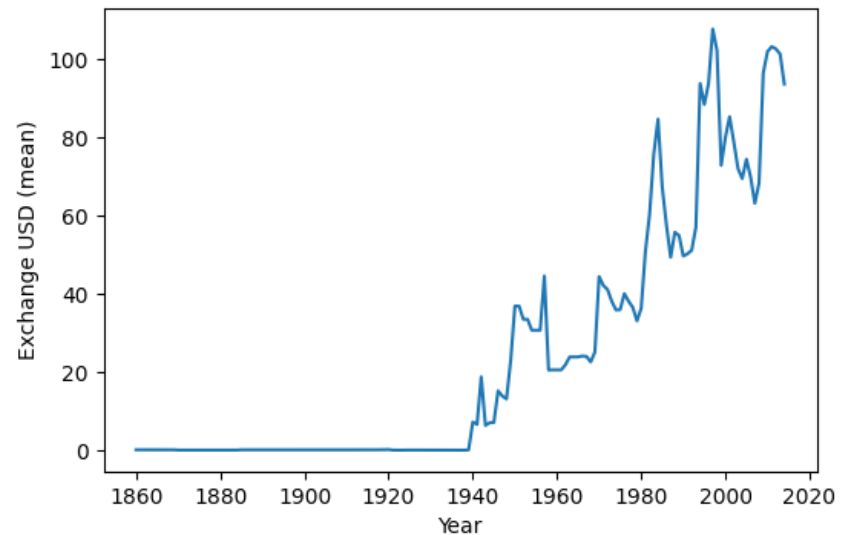
Observe el gráfico que obtuvimos ¿Qué puede deducir de este gráfico?

Evolución del tipo de cambio desde 1860 a 2014



plt.scatter

Evolución del tipo de cambio desde 1860 a 2014



plt.plot

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('african_crises.csv')

data = df[(df['banking_crisis'] == 'no_crisis') &
          (df['year'] > 1940) &
          (df['year'] < 1980)]

exchange = data['inflation_annual_cpi'].dropna()

plt.hist(exchange, bins=50)
```

dropna() elimina los registros que no tienen dato de Altura

hist() nos permite graficar histogramas (agrupación por rango de índice de precios al consumidor en este caso)


```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('african_crises.csv')

data = df[(df['banking_crisis']=='no_crisis') &
          (df['year']>1940) &
          (df['year']<1980)]

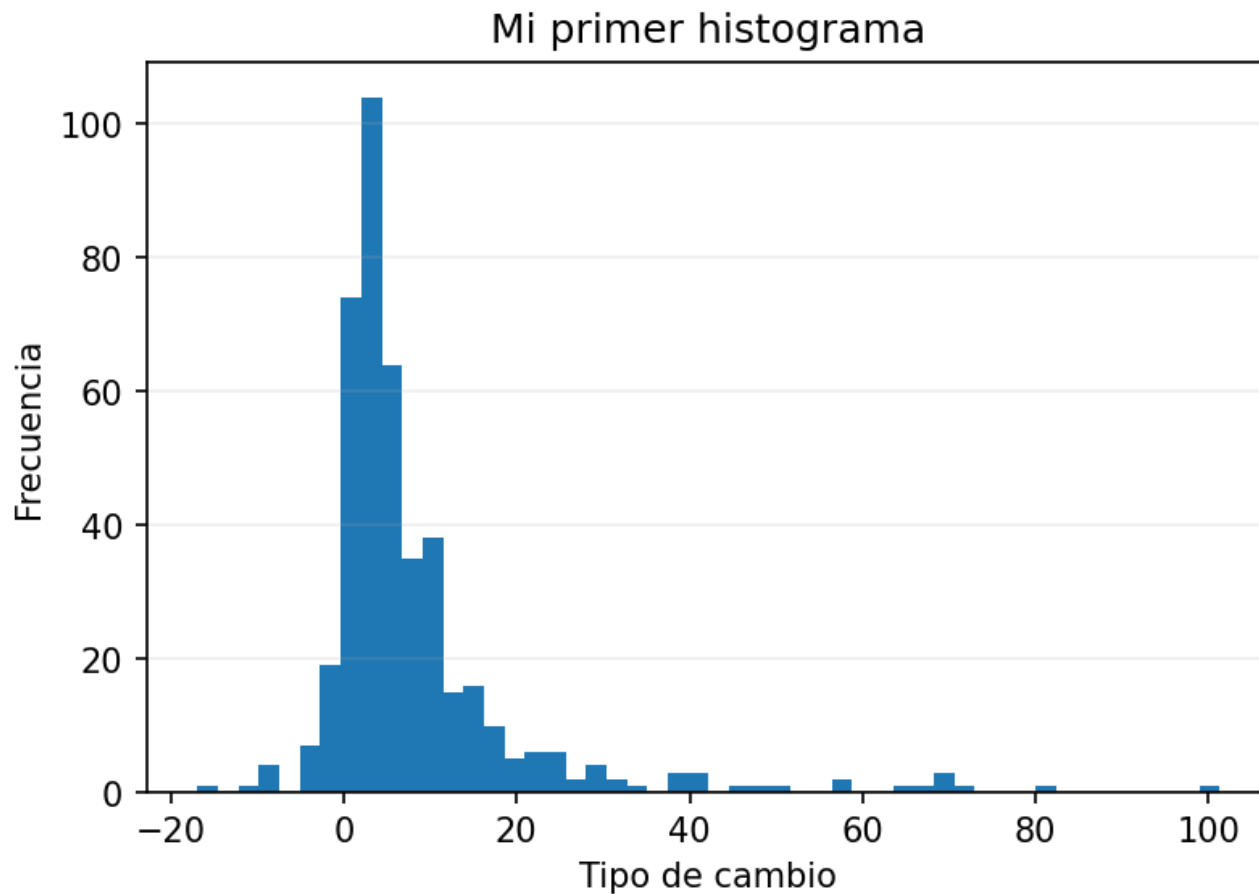
exchange= data['inflation_annual_cpi'].dropna()

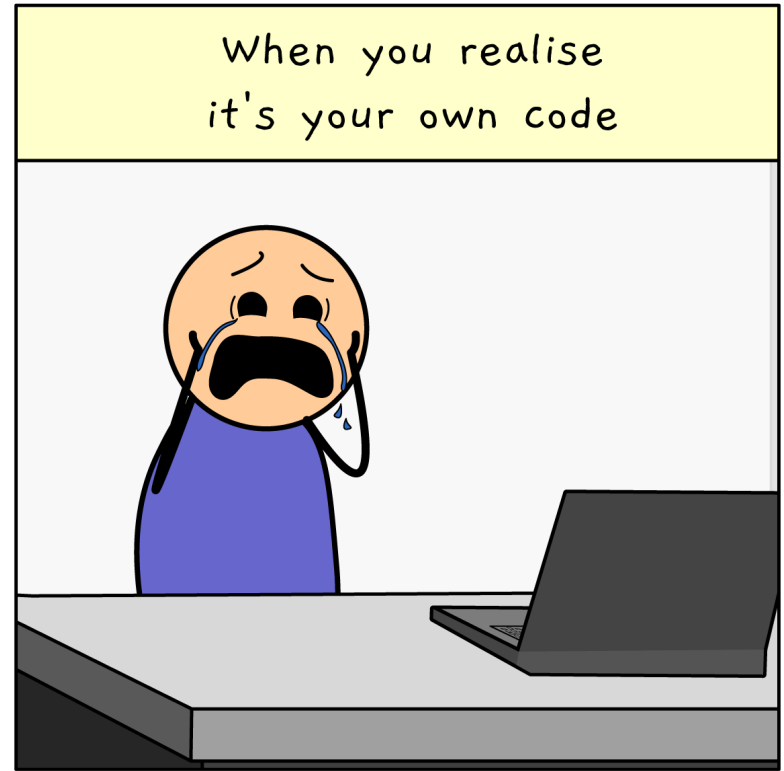
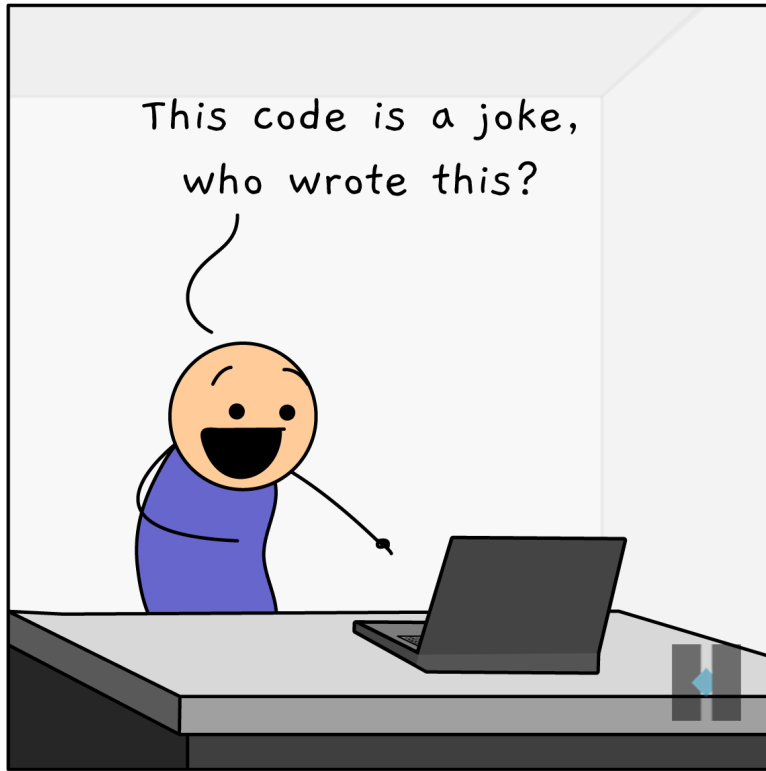
plt.hist(exchange, bins=50)
plt.xlabel('Tipo de cambio')
plt.ylabel('Frecuencia')
plt.title('Mi primer histograma')
plt.grid(axis='y',alpha=0.2)
plt.show()
```

Opciones adicionales para complementar el gráfico



Observe el gráfico que obtuvimos ¿Qué puede deducir de este gráfico?



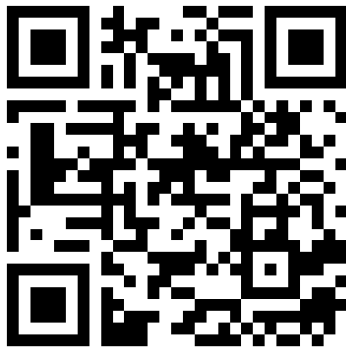


f /techindustan

🐦 /techindustan

📷 /techindustan

Responde
este
cuestionario
desde este
link 🖱️



Preguntas curiosas. [Python]

Por favor contesta de la forma seria y responsable. Los resultados son completamente anónimos

***Obligatorio**

¿Eres feliz? *

Elige ▼

Indica tu sexo *

- Mujer
- Hombre
- Prefiero no decirlo

Con escala de 1 a 7, ¿qué nota asignarías a la facilidad para programar con Python? *

<https://forms.gle/PoMVfj7k3GL9bZpT7>

```
import pandas as pd

data = pd.pandas.read_csv('Formulario.csv')
print(data.columns) ←
```

Podemos conocer las columnas de los datos, sin tener que abrir el archivo CSV

Index	Marca temporal	¿Eres feliz?	Indica tu sexo	¿signarías a la facilidad?	que más te acomoda	más apropiado para	que se utiliza más	10 ¿cuán agotado?	¿qué te ha parecido?
0	2019/12/...	No	Hombre	1	En la ta...	R	Python	7	4
1	2019/12/...	Sí	nan	5	En la ma...	Python	R	9	10

Format Resize Background color Column min/max Save and Close Close

```
import pandas as pd

data = pd.pandas.read_csv('Formulario.csv')
print(data.columns)

score_python = np.array(data[data.columns[3]])
score_agotado = np.array(data[data.columns[7]])
general_score = np.array(data[data.columns[8]])
```

No es necesario indicar el nombre directamente, lo podemos obtener a través del índice de las columnas

En caso que sea necesario, podemos transformar los datos de tipo Dataframe a un arreglo de tipo numpy

```
import pandas as pd

data = pd.pandas.read_csv('Formulario.csv')
print(data.columns)

score_python      = np.array(data[data.columns[3]])
score_agotado     = np.array(data[data.columns[7]])
general_score     = np.array(data[data.columns[8]])

score_python      = score_python[:, np.newaxis]
score_agotado     = score_agotado[:, np.newaxis]
general_score     = general_score[:, np.newaxis]
```

Esta operación nos permite agregar una dimensión al problema. Pasamos de una lista a una matriz

```
import pandas as pd

data = pd.pandas.read_csv('Formulario.csv')
print(data.columns)

score_python      = np.array(data[data.columns[3]])
score_agotado     = np.array(data[data.columns[7]])
general_score     = np.array(data[data.columns[8]])

score_python      = score_python[:, np.newaxis]
score_agotado     = score_agotado[:, np.newaxis]
general_score     = general_score[:, np.newaxis]

print(data.columns[3], 'promedio:', score_python.mean())
print(data.columns[7], 'promedio:', score_agotado.mean())
print(data.columns[8], 'promedio:', general_score.mean())
```

Dado que los datos son de tipo numpy, podemos emplear todas las herramientas que ésta posee



Tiempo : 10 minutos

1. Genera un histograma por cada uno de los puntajes anteriores empleando la biblioteca Matplotlib
 - `score_python,`
 - `score_agotado,`
 - `general_score.`

- ▶ Introducción a Pandas
 - Operaciones básicas de datos
 - Agrupaciones de datos
 - Pivot table
 - CrossTab
 - Gráfica matplotlib
 - Gráfica Plotly



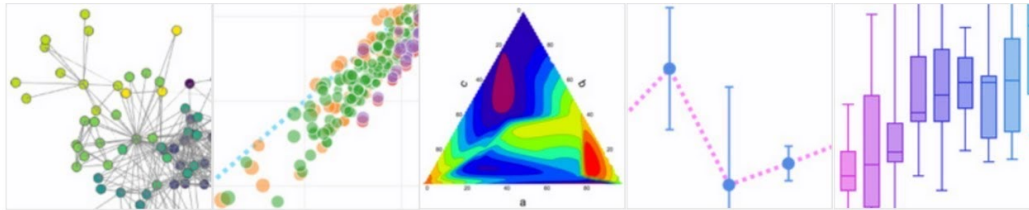
```
self.FidValue = OrderedDict(sorted(self.items(), key=lambda item: item[0]))
#Read item in dictionary
for key, value in item.FidValue.items():
    typeOfFID = mapFidType[key]
    if (typeOfFID == "DATE"):
        d = datetime.datetime.strptime(str(value), "%Y-%m-%d")
        dataCal = datetime.date.strptime(str(d), "%Y-%m-%d")
        FidAndValue = FidAndValue + value
    else: FidAndValue = FidAndValue + value
```

```
try:
    start = date(int(self.start_year.get(0)),
                self.months.index(self.start_month.get(0)),
                int(self.start_day.get(0)))
    end = date(int(self.end_year.get(0)),
              self.months.index(self.end_month.get(0)),
              int(self.end_day.get(0)))
```

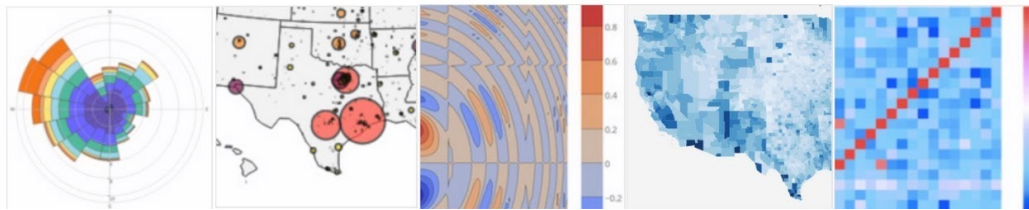


Plotly es una biblioteca especializada en el gráfico de datos (en la web). Para emplearla, debemos instalarla en nuestro ambiente de trabajo

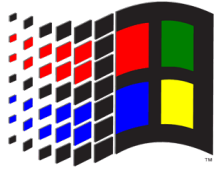
```
pip install plotly
```



LEARN PLOTLY



```
conda install -c plotly plotly spyder
```



En Windows abrir **Anaconda Prompt** y escribir:

```
conda install -c plotly plotly spyder
```

The screenshot shows the Anaconda Navigator application window. In the foreground, a terminal window is open with the command `conda install -c plotly plotly spyder` entered. A dashed orange box highlights the command, and a yellow callout box with an arrow points to it, containing the text "escribe el comando y presiona enter". The background shows the Anaconda Navigator interface with a sidebar on the left and a search bar on the right.



En MacOSX abrir una terminal y escribir:

```
mlacarrasco — -bash — 80x24
(base) mbpdemiguel721:~ mlacarrasco$ conda install -c plotly plotly spyder

Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

(base) mbpdemiguel721:~ mlacarrasco$
(base) mbpdemiguel721:~ mlacarrasco$
```

```
from plotly.offline import plot
import plotly.express as px
import pandas as pd

df = pd.read_csv('african_crises.csv')
data = df[(df['banking_crisis']=='no_crisis') &
          (df['year']>1940) &
          (df['year']<1980)]

exchange= data['inflation_annual_cpi'].dropna()

fig = px.histogram(exchange, x="inflation_annual_cpi")

plot(fig, auto_open=True)
```

histogram, crea un histograma. Según los valores de x, y, se genera un histograma. Los bins son estimados según los datos

abre una Ventana en un browser para general el gráfico. Este gráfico es off-line, pero se puede generar un gráfico online con una cuenta.

```
import pandas as pd
import plotly.express as px
import plotly
```

```
plotly.io.renderers.default = 'colab' ←
```

```
import plotly.graph_objects as go
df = pd.read_csv('african_crises.csv')
```

```
data = df[(df['banking_crisis']=='no_crisis') &
          (df['year']>1940) &
          (df['year']<1980)]
```

```
exchange = data['inflation_annual_cpi'].dropna()
fig = px.histogram(exchange, x="inflation_annual_cpi") ←
```

```
fig.show() ←
```

Genera el gráfico
dentro del ambiente
de Colab

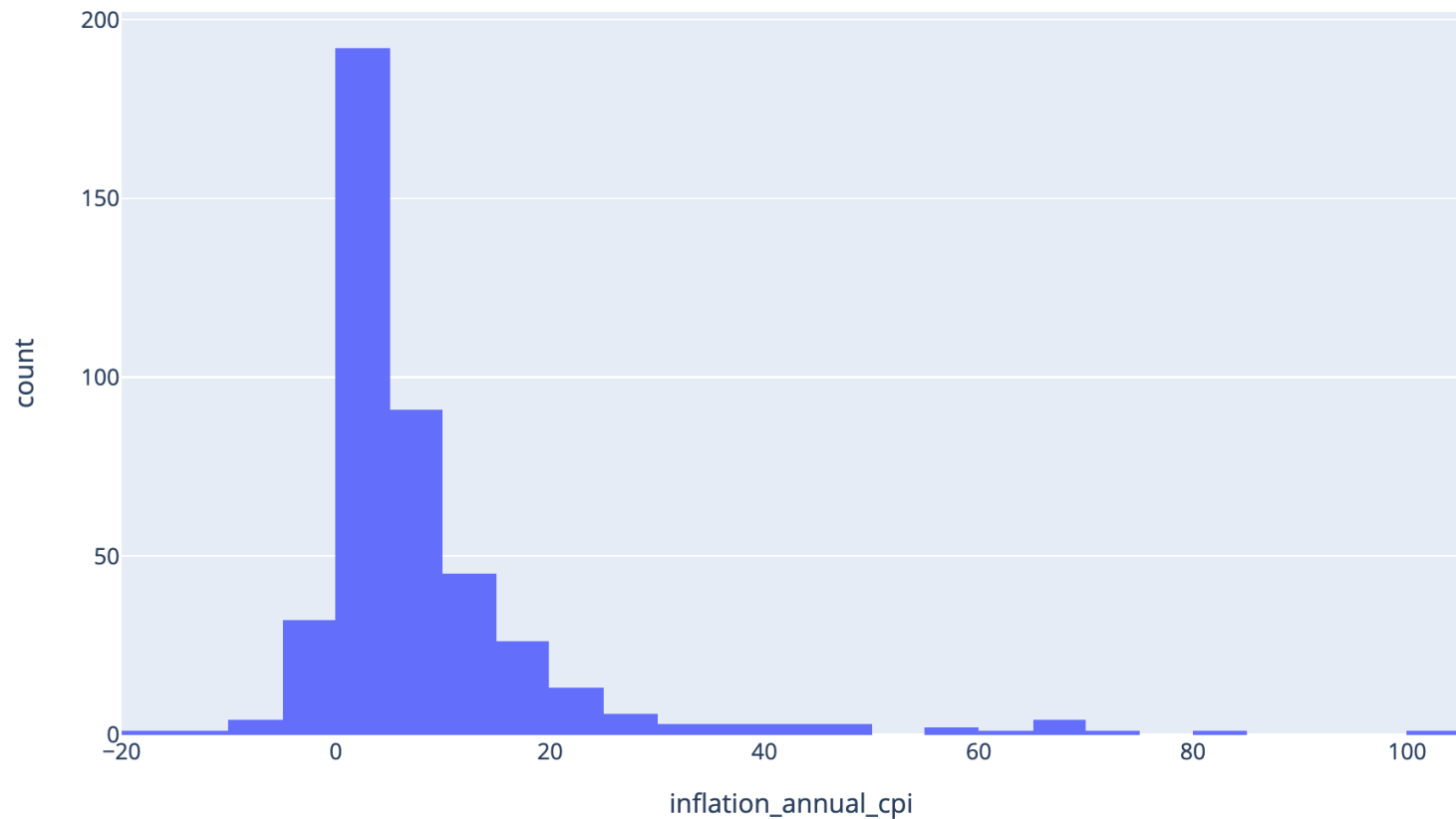
The logo for Google Colab, featuring the word "Google" in its multi-colored font above the word "colab" in a bold, orange, lowercase sans-serif font.

histogram, crea un histograma.
Según los valores de x, y, se
genera un histograma. Los bins
son estimados según los datos



Observe el gráfico que obtuvimos ¿Qué puede deducir de este gráfico?

Distribución de índices de precio al consumidor de países Africanos entre 1940 y 1980




```
import pandas as pd
import plotly.express as px
import plotly

df = pd.read_csv('african_crises.csv')

data = df[(df['banking_crisis']=='no_crisis')]

exchange= data[['inflation_annual_cpi', 'exch_usd']].dropna()

fig = px.scatter(exchange, x="inflation_annual_cpi", y="exch_usd")

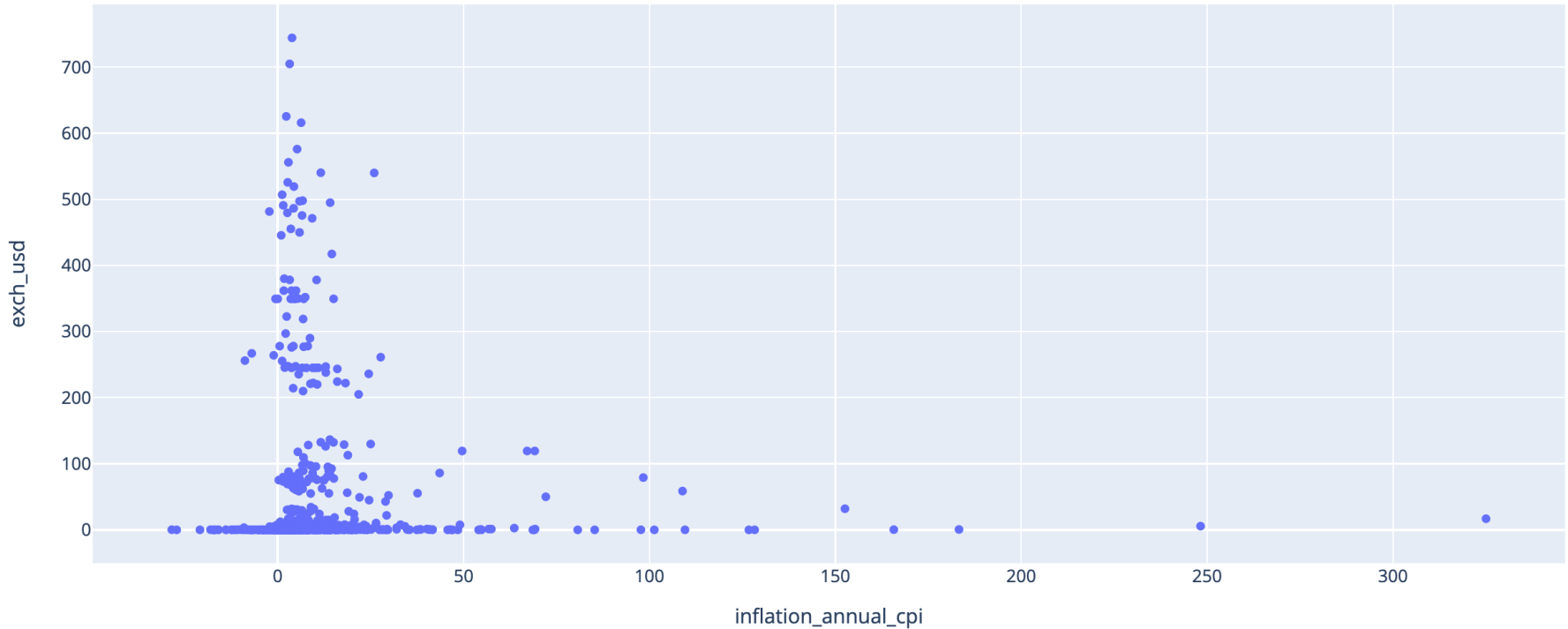
fig.show()
```

scatter crea un gráfico de dispersión de puntos en los ejes X,Y.



Observe el gráfico que obtuvimos ¿Qué puede deducir de este gráfico?

Gráfico de dispersión de inflación y tipo de cambio



```
import pandas as pd
import plotly.express as px
import plotly

plotly.io.renderers.default = 'colab'

import plotly.graph_objects as go

df = pd.read_csv('african_crises.csv')

grupo = df.groupby('year').agg(['mean'])

dX = grupo.index
dY = grupo['exch_usd']['mean']

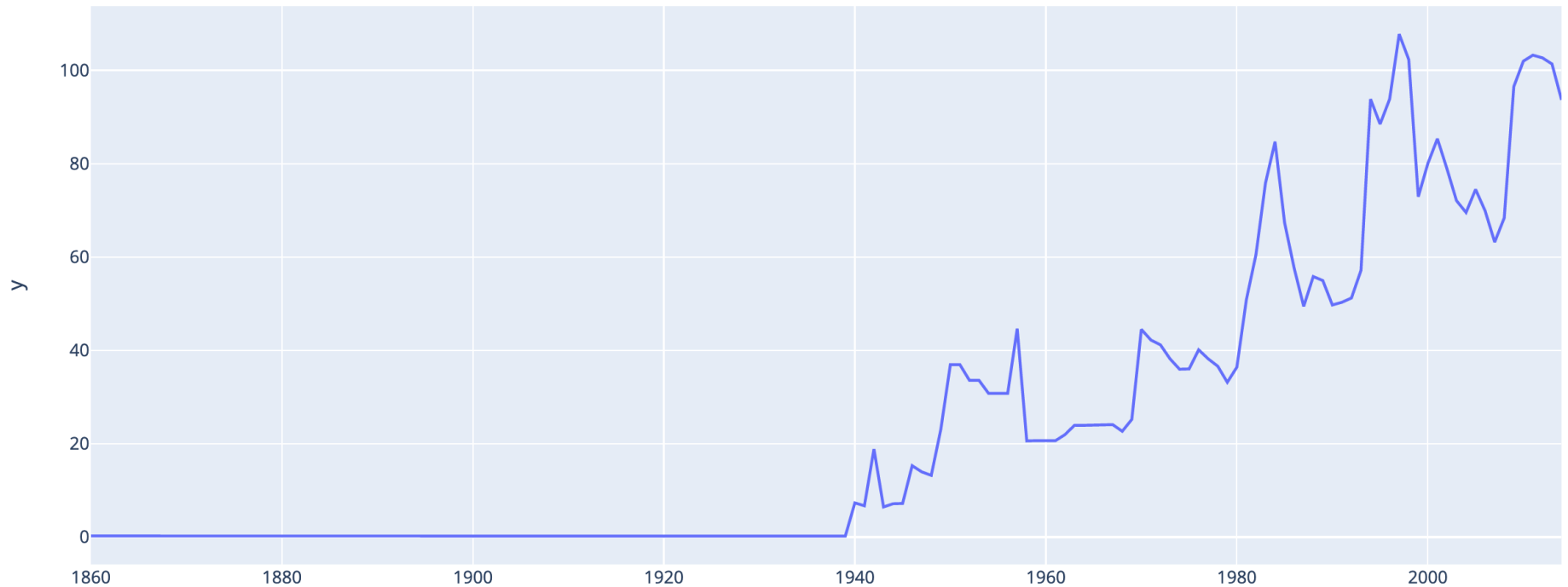
fig = px.line(x=dX, y=dY)
fig.show()
```

Podemos asignar los
datos de cada eje a
una variable



Observe el gráfico que obtuvimos ¿Qué puede deducir de este gráfico?

Evolución del tipo de cambio desde
1860 a 2014



<https://elitedatascience.com/r-vs-python-for-data-science>

The screenshot shows the top of a webpage. On the left is the 'ELITE DATA SCIENCE' logo. On the right are navigation links for 'DATA SCIENCE PRIMER', 'ARTICLES', and a search icon. Below the navigation is a dark banner with the text 'Free Download: DS Career Guide' and a 'GET INSTANT ACCESS!' button. The main content area features a dark background with a large Python logo and a smaller R logo. The title 'R vs. Python for Data Science: Summary of Modern Advances' is centered in white. A 'EXPLAINERS' tag is above the title. On the left side of the article area is a vertical social media sharing bar with icons for LinkedIn, Facebook, Instagram, Twitter, Reddit, Email, and Print. At the bottom left of the screenshot is the URL 'https://elitedatascience.com'.



Seriously why?