

INTRODUCCIÓN A PYTHON

INTRODUCCIÓN

2026

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



```

self.FidValue = OrderedDict(sorted(self.items(),
#Read item in dictionary
for key, value in item.FidValue.items():
    typeOfFID = mapFidType.get(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)),
                int(self.start_day.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)),
            int(self.end_day.get(
                self.months.index(self.end_month))
            )
        )
    )

```

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

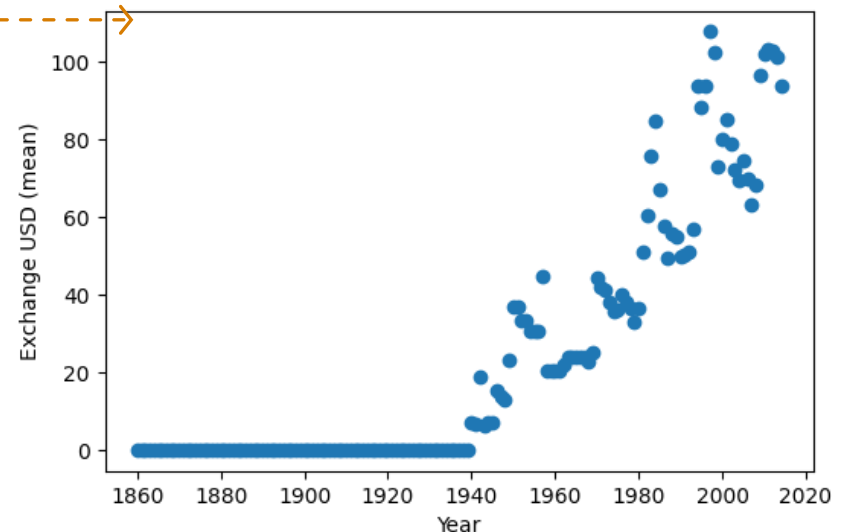
df = pd.read_csv('african_crises.csv')
print(df.columns)

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)

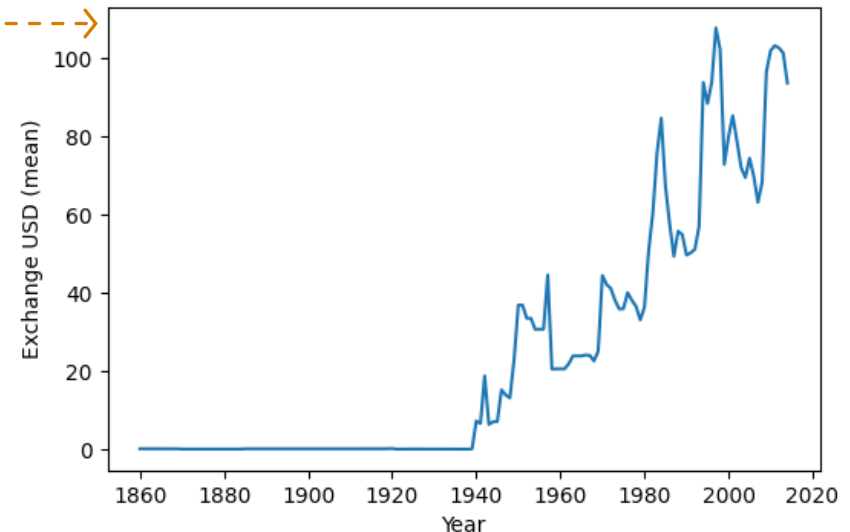
group_1 = df.groupby(['year'])
stats_1 = group_1['exch_usd'].agg(['mean', 'std'])

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

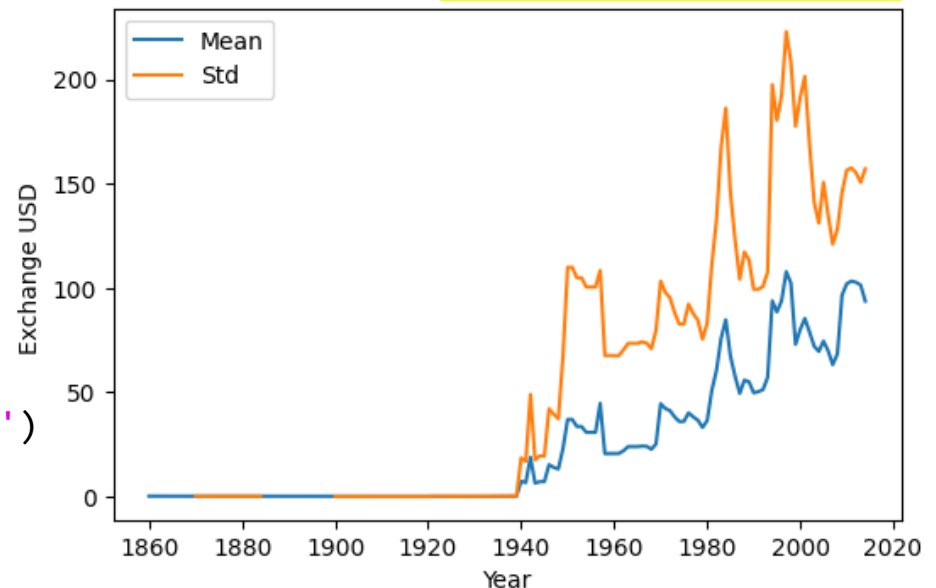
df = pd.read_csv('african_crises.csv')
print(df.columns)
group_1 = df.groupby(['year'])
stats_1 = group_1['exch_usd'].agg(['mean', 'std'])
```

```
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')

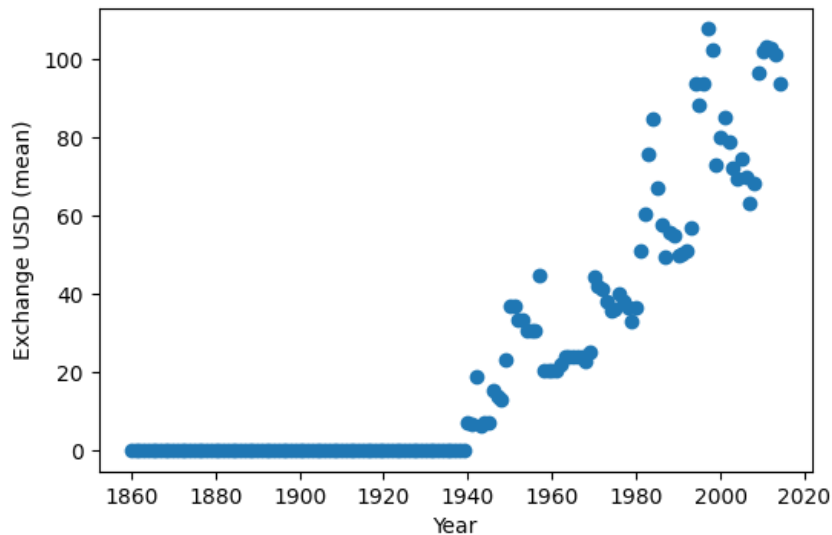
plt.show()
```

savefig() permite guardar el gráfico en un archivo



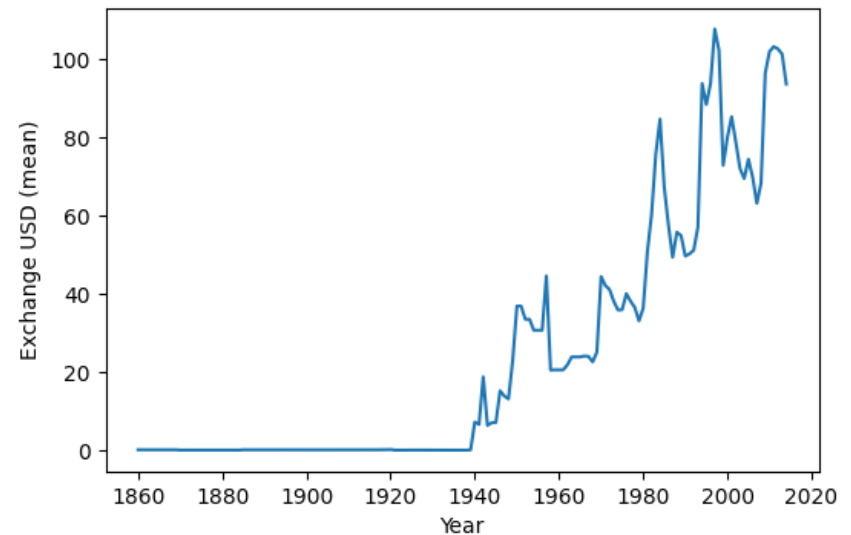
Observe el gráfico que obtuvimos ¿Qué puede deducir de este gráfico?

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)]

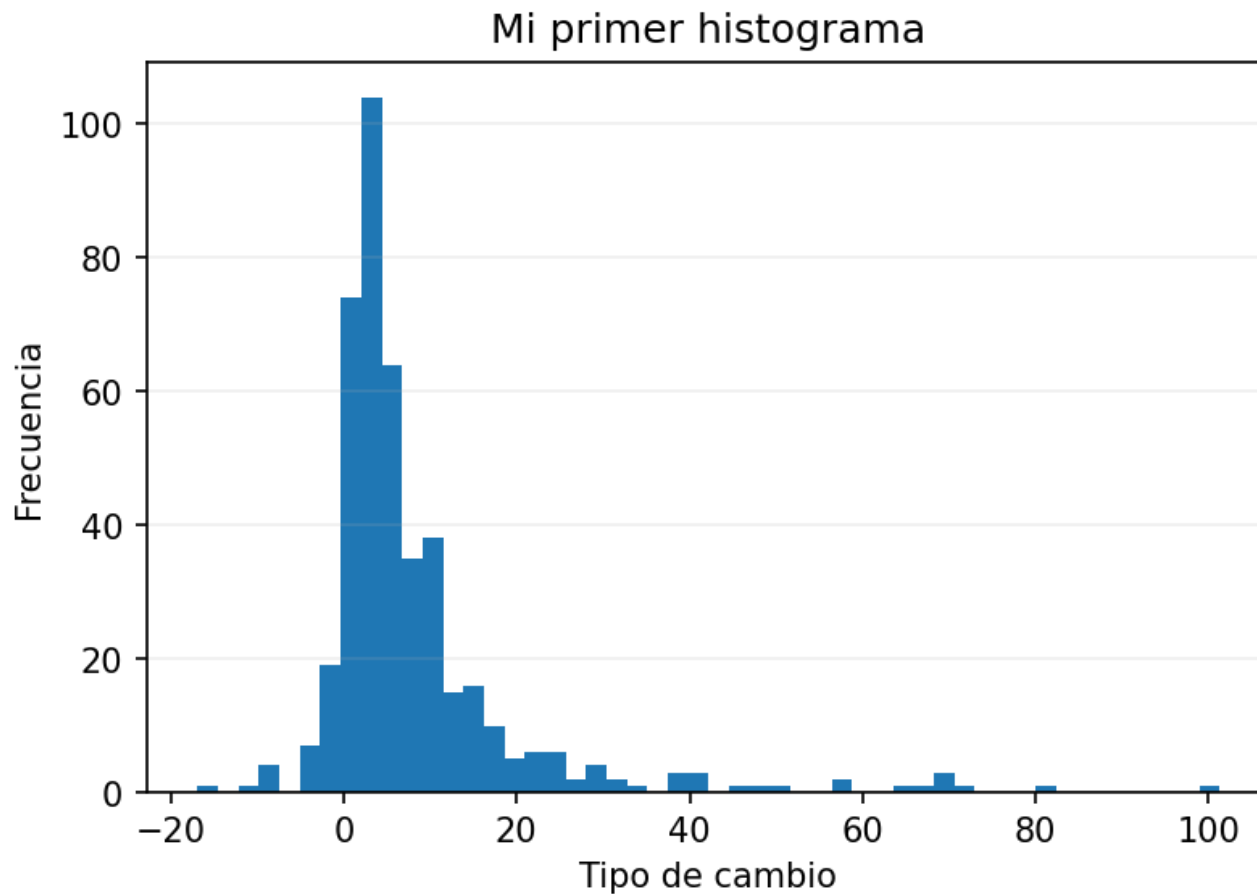
exchange= data['inflation_annual_cpi'].dropna()

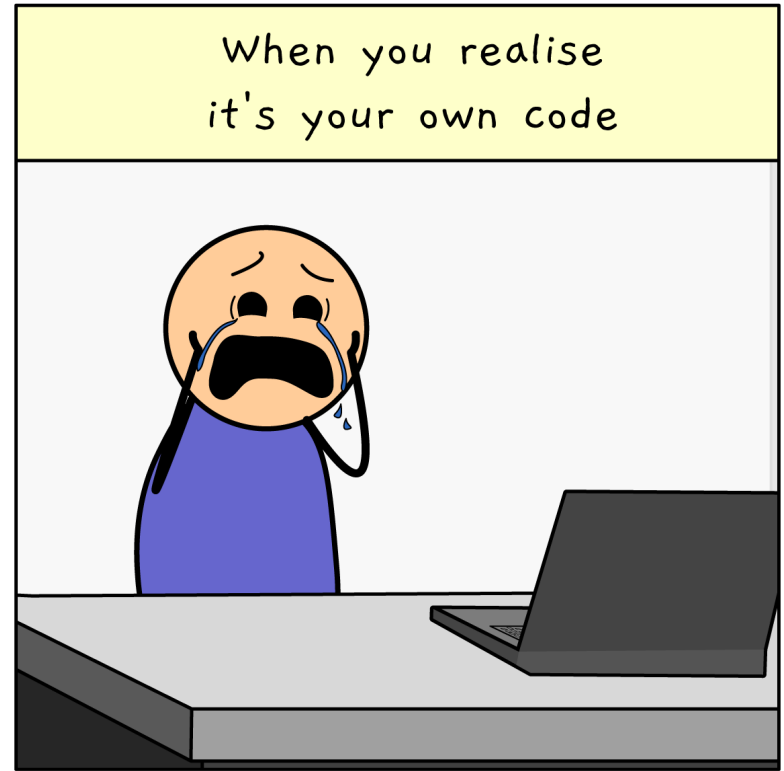
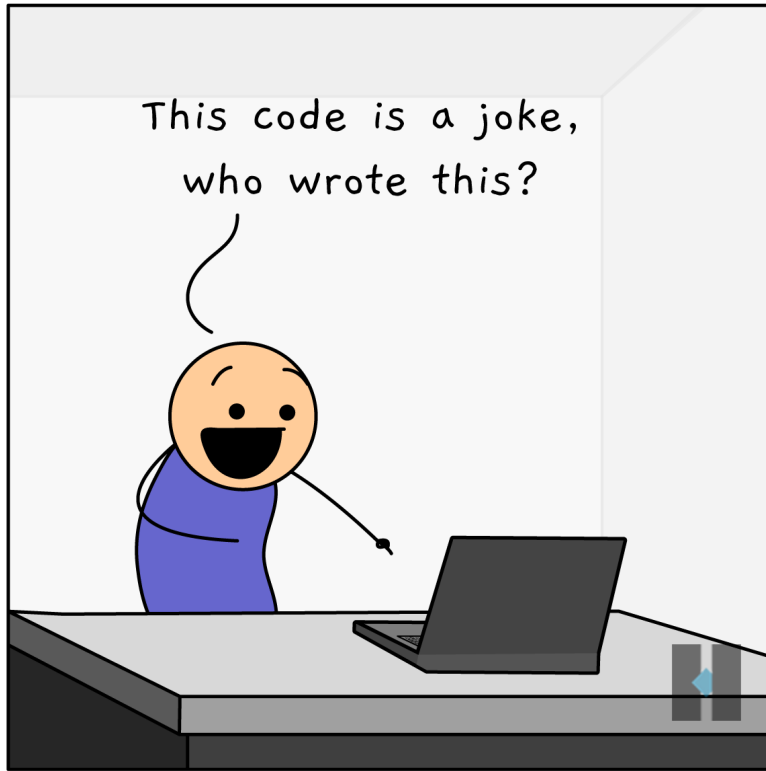
plt.hist(exchange, bins=50)
plt.xlabel('Tipo de cambio')
plt.ylabel('Frecuencia')
plt.title('Mi primer histograma')
plt.grid(axis='y',alpha=0.2)
plt.show()
```

Opciones adicionales para complementar el gráfico



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





f /techindustan

🐦 /techindustan

📷 /techindustan

- ▶ 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.get(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
    
```

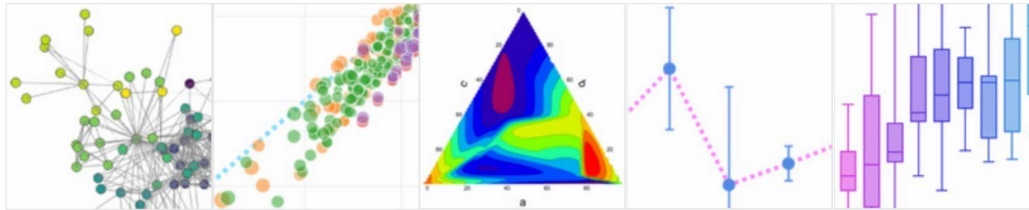
```

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

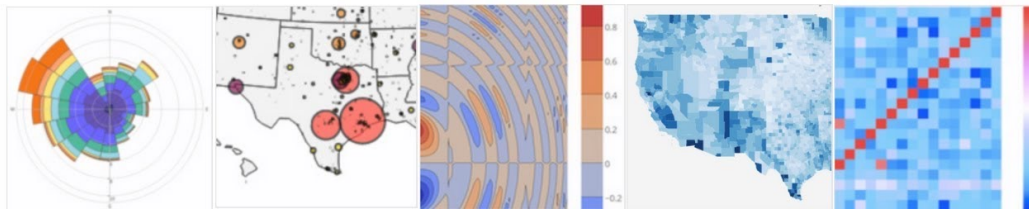


Plotly es una biblioteca especializada en el gráfico de datos (en la web). Para emplearla, debemos instalarla en nuestro ambiente de trabajo

```
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 the text "escribe el comando y presiona enter" points to it. 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:

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

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

# All requested packages already installed.

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

```
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() ←
```

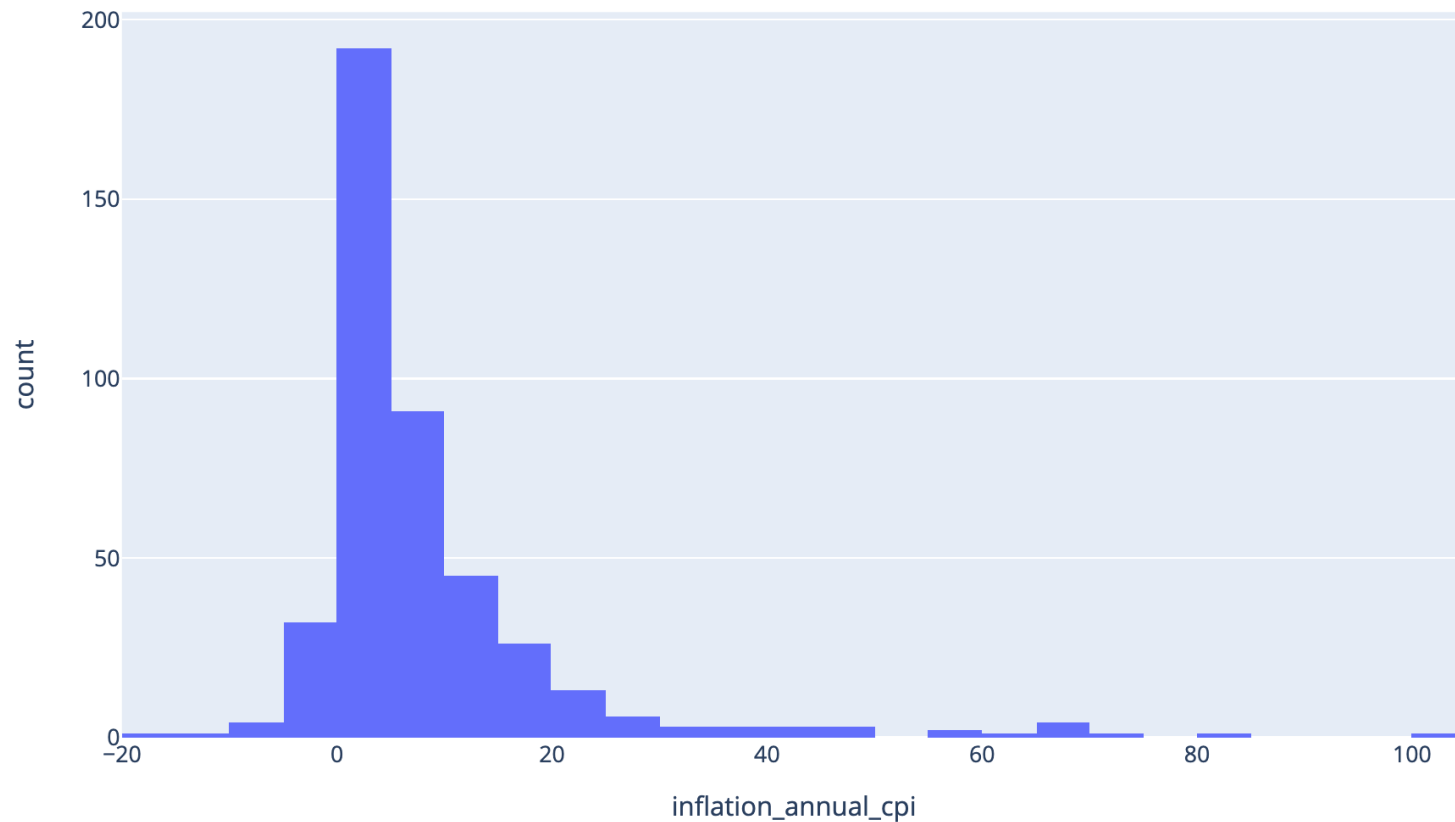
Genera el gráfico
dentro del ambiente
de Colab

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")

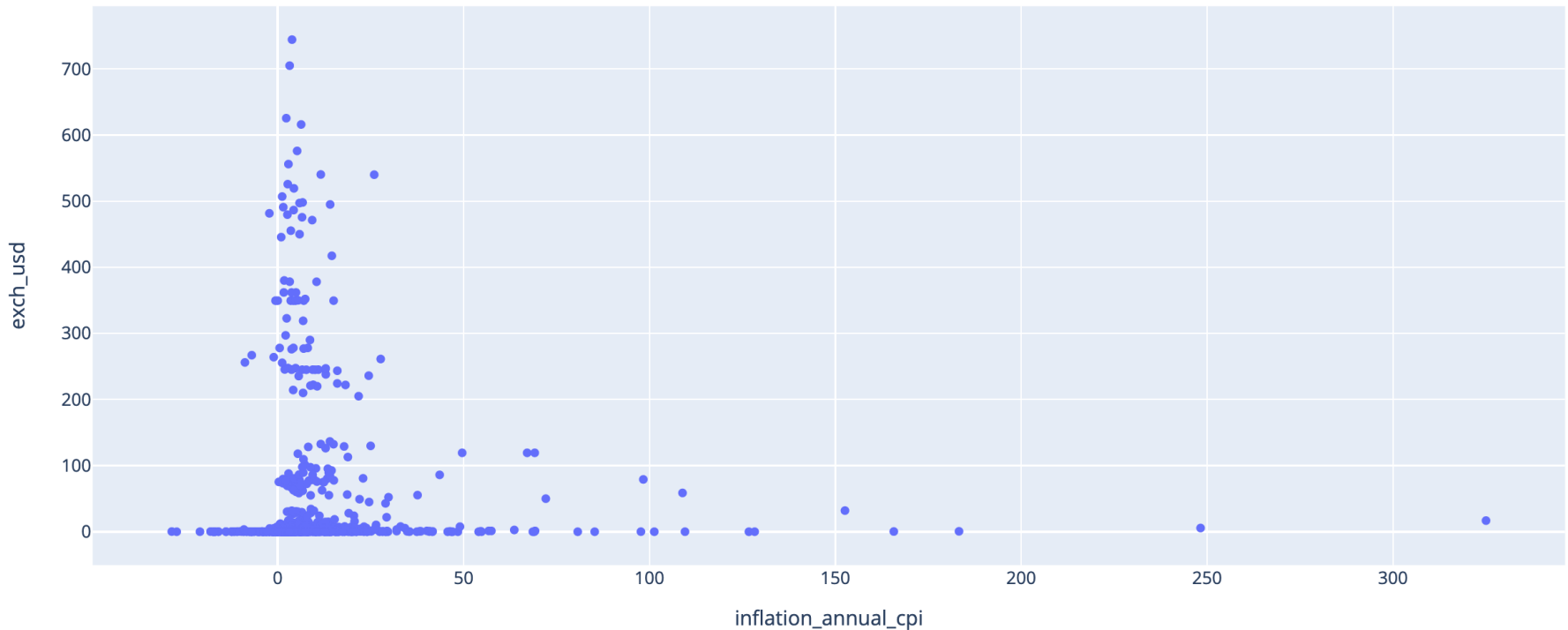
fig.show()
```

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']

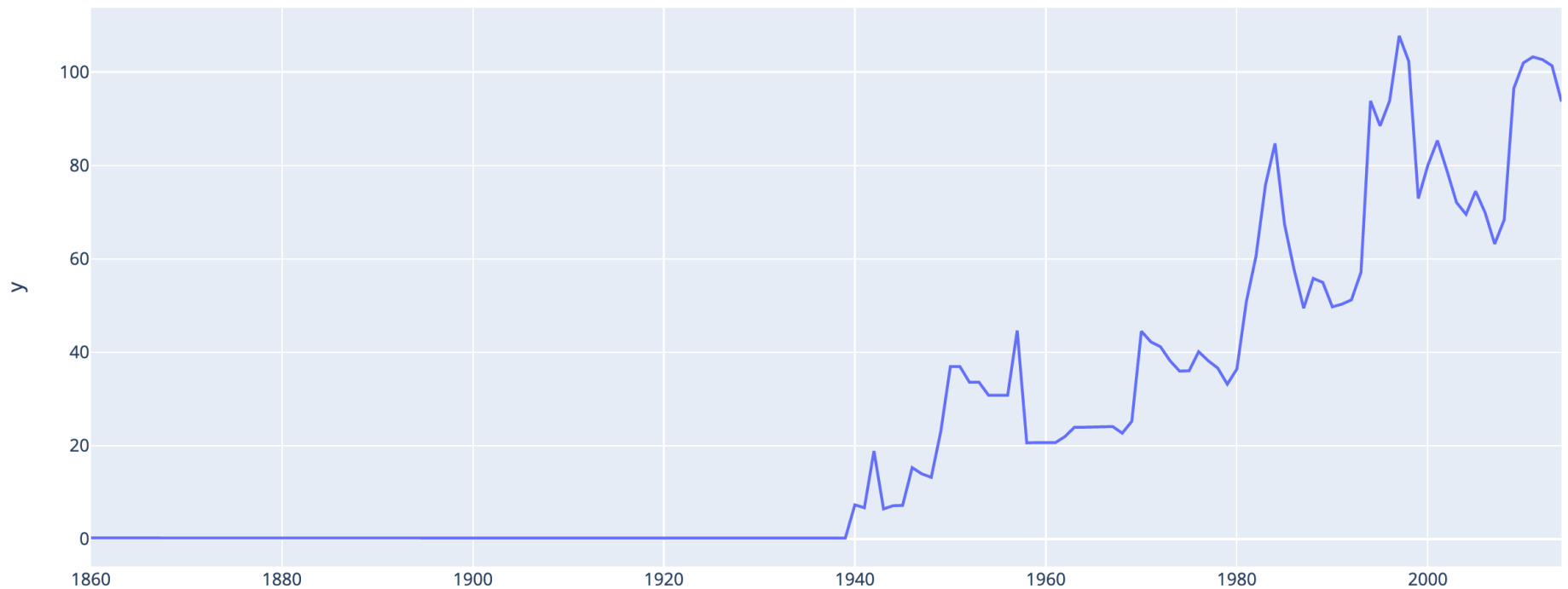
fig = px.line(x=dX, y=dY)
fig.show()
```

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 a webpage from elite-datascience.com. At the top left is the logo for 'ELITE DATA SCIENCE'. On the right side of the top navigation bar are links for 'DATA SCIENCE PRIMER', 'ARTICLES', and a search icon. Below the navigation bar is a dark banner with the text 'Free Download: DS Career Guide' and a sub-headline 'How to Learn Data Science & Machine Learning, Land a High-Paying Job, and Future-Proof Your Career'. A blue button on the right of the banner says 'GET INSTANT ACCESS!'. The main content area features a dark background with a large Python logo and a large 'R' logo. The title 'R vs. Python for Data Science: Summary of Modern Advances' is centered in white text, with a green 'EXPLAINERS' tag above it. On the left side of the article area is a vertical stack of social media sharing icons: LinkedIn, Facebook, Google+, Twitter, Reddit, Email, and Print. At the bottom left of the page, the URL 'https://elitedatascience.com' is visible.



Seriously why?