

TEMA	PREGUNTA	RESPUESTA	JUSTIFICACION
Tamaño del Dataset	Cuántas filas y columnas tiene?	El archivo tiene 117 filas y 13 columnas.	
	Es suficiente para el análisis?	Depende del objetivo del análisis	El dataset tiene varios años de datos sobre quejas y pérdidas económicas, lo que podría ser útil.
Formato de los datos	Esta en CSV, JSON, SQL u otro formato?	Esta en formato CSV	
	Es fácil de procesar?	SI, porque tiene tipos de datos claros	Tiene 12 columnas numéricas (int64) y 1 columna con nombres de países (object).
Etiquetado	Los datos tienen etiquetas claras?	SI	Los nombres de las columnas indican el año y el tipo de dato.
	Hay valores faltantes o datos inconsistentes?	SI	Hay 1 valor nulo en la columna "Country". No hay valores nulos en las demás columnas.
Fuente y credibilidad	¿Quién creó el dataset?	Lo creó un estudiante de la Universidad de Wuhan (WHU) en Wuhan, llamado HUZPSB	Este dataset contiene información sobre el número de informes de incidentes de ciberseguridad presentados a las autoridades locales y las pérdidas estimadas en dólares estadounidenses.
	¿Está bien documentado?	Está bien documentado	Con descripciones claras y recursos adicionales proporcionados por el autor para facilitar su uso y comprensión

# RESULTADOS

The screenshot shows a Kaggle notebook titled 'notebookee1e15d6ba'. The interface includes a top navigation bar with links to various services, a left sidebar with icons for file management and execution, and a right sidebar with notebook controls. The main area contains two code cells. The first cell is a comment block explaining the environment and providing instructions on how to use the notebook. The second cell contains Python code that imports pandas and sets up the file path for a CSV file named 'LossFromNetCrime.csv'.

```
[2]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session

/kaggle/input/cybersecurity-incidents-dataset/LossFromNetCrime.csv

+ Code + Markdown

import pandas as pd

# Ruta del archivo en Kaggle
file_path = "/kaggle/input/cybersecurity-incidents-dataset/LossFromNetCrime.csv"

# Lee el archivo CSV
df = pd.read_csv(file_path)

# Obtiene información general del dataset
df.info()

# Muestra el número de filas y columnas
num_filas, num_columnas = df.shape
print(f'El archivo tiene {num_filas} filas y {num_columnas} columnas.')
```

The screenshot shows the same Kaggle notebook interface, but with the code cells executed. The first cell is now a comment block. The second cell contains Python code that imports pandas and sets up the file path for a CSV file named 'LossFromNetCrime.csv'. The third cell contains Python code that prints the first few rows of the dataset. The fourth cell contains Python code that prints the column names of the dataset. The fifth cell contains Python code that prints the number of non-null values in each column. The sixth cell contains Python code that prints the unique values in each column.

```
+ Code + Markdown

import pandas as pd

# Ruta del archivo en Kaggle
file_path = "/kaggle/input/cybersecurity-incidents-dataset/LossFromNetCrime.csv"

# Lee el archivo CSV
df = pd.read_csv(file_path)

# Obtiene información general del dataset
df.info()

# Muestra el número de filas y columnas
num_filas, num_columnas = df.shape
print(f'El archivo tiene {num_filas} filas y {num_columnas} columnas.')
```

```
[ ]: #Revisa las primeras filas
print(df.head())
```

```
[ ]: #Muestra los nombres de columnas
#df.columns
print(df.columns)
```

```
[ ]: #Cuenta los valores nulos en cada columna
print(df.isnull().sum())
```

```
[ ]: #Muestra valores únicos en cada columna
print(df.nunique())
```

notebookee1e15d6ba

```
import pandas as pd

# Ruta del archivo en Kaggle
file_path = "/kaggle/input/cybersecurity-incidents-dataset/LossFromNetCrime.csv"

# Lee el archivo CSV
df = pd.read_csv(file_path)

# Obtiene información general del dataset
df.info()

# Muestra el número de filas y columnas
num_filas, num_columnas = df.shape
print(f"El archivo tiene {num_filas} filas y {num_columnas} columnas.")
```

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 117 entries, 0 to 116  
Data columns (total 13 columns):  
# Column Non-Null Count Dtype  
--- ---  
0 Country 116 non-null object  
1 2019\_Complaints 117 non-null int64  
2 2019\_Losses 117 non-null int64  
3 2020\_Complaints 117 non-null int64  
4 2020\_Losses 117 non-null int64  
5 2021\_Complaints 117 non-null int64  
6 2021\_Losses 117 non-null int64  
7 2022\_Complaints 117 non-null int64  
8 2022\_Losses 117 non-null int64  
9 2023\_Complaints 117 non-null int64  
10 2023\_Losses 117 non-null int64  
11 2024\_Complaints 117 non-null int64  
12 2024\_Losses 117 non-null int64  
dtypes: int64(12), object(1)  
memory usage: 12.0+ KB  
El archivo tiene 117 filas y 13 columnas.

+ Code + Markdown

notebookee1e15d6ba

```
[2]: #Revisa las primeras filas
print(df.head())
```

	Country	2019_Complaints	2019_Losses	2020_Complaints	2020_Losses
0	PR	655	5929974	1338	7209755
1	PS	1784	22483591	2890	25423219
2	PT	1119	13870074	2020	12391290
3	PY	1913	10967865	2992	13815152
4	AE	5503	48101706	7390	81178182

  

	2021_Complaints	2021_Losses	2022_Complaints	2022_Losses
0	1785	9463928	1594	17211758
1	3352	48918464	3210	57796003
2	2102	18205913	1918	30859319
3	3188	26677869	3768	40114594
4	10164	131934602	10042	187062157

  

	2023_Complaints	2023_Losses	2024_Complaints	2024_Losses
0	1817	21018302	1974	31545772
1	3378	69292891	3811	66002407
2	2178	28700418	2209	40192274
3	3487	33579643	2678	45168957
4	11034	244238485	12071	280815215

```
#Muestra los nombres de columnas
#df.columns
print(df.columns)
```

Index(['Country', '2019\_Complaints', '2019\_Losses', '2020\_Complaints',  
'2020\_Losses', '2021\_Complaints', '2021\_Losses', '2022\_Complaints',  
'2022\_Losses', '2023\_Complaints', '2023\_Losses', '2024\_Complaints',  
'2024\_Losses'],  
dtype='object')

+ Code + Markdown

←

→

↺

🏠

🔒 https://www.kaggle.com/code/luispachecoalvarado/notebookee1e15d6ba/edit

🔍 90% 🔌 4°

Hotmail.com Network Control Login DNS Hostinger Hostinger Webmail EcuaWeb LACNIC Campus LACNIC Udemey ChatGPT BLACKBOXAI DeepSeek - Into the U... WhatsApp In

notebookee1e15d6ba

File Edit View Run Settings Add-ons Help

+

+

✂

📄

📄

▶▶▶ Run All

Code

Draft Session (1m)

🔌 ⏻ ↺ ⋮

[4]:

#Cuenta los valores nulos en cada columna

print(df.isnull().sum())

Country 1

2019\_Complaints 0

2019\_Losses 0

2020\_Complaints 0

2020\_Losses 0

2021\_Complaints 0

2021\_Losses 0

2022\_Complaints 0

2022\_Losses 0

2023\_Complaints 0

2023\_Losses 0

2024\_Complaints 0

2024\_Losses 0

dtype: int64

▶

#Muestra valores únicos en cada columna

print(df.nunique())

Country 116

2019\_Complaints 116

2019\_Losses 117

2020\_Complaints 117

2020\_Losses 117

2021\_Complaints 116

2021\_Losses 117

2022\_Complaints 115

2022\_Losses 117

2023\_Complaints 116

2023\_Losses 117

2024\_Complaints 117

2024\_Losses 117

dtype: int64

+ Code

+ Markdown