PECIER

Comité Nacional Peruano de la CIER

Curso Virtual

Python para el Análisis de Datos y la Automatización en el Sector Eléctrico

27, 29 y 31 de Octubre, 03, 05, 07, 10 y 12 de Noviembre 2025.

MARVIN COTO – FACILITADOR
E-mail: mcotoj@gmail.com

PECIER

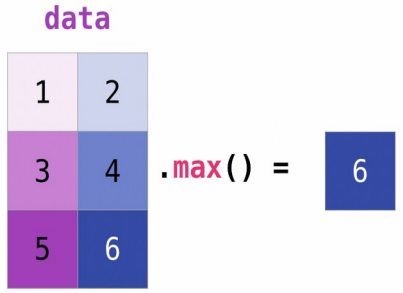
Comité Nacional Peruano de la CIER

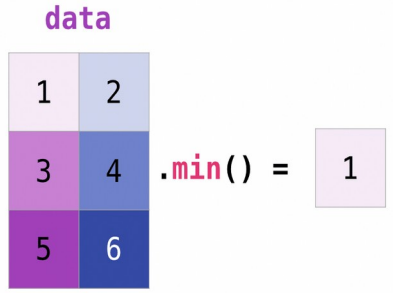
SESIÓN 5
Parte 1

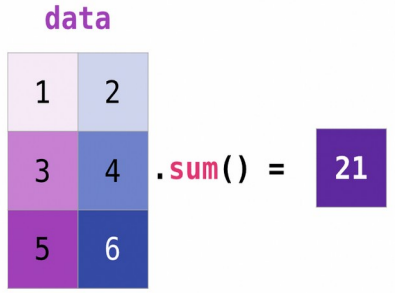
PECIER
Comité Nacional Peruano de la CIER

En la sesión anterior...

Numpy: Manejo de arreglos y eficiencia en cálculos.







PECIER
Comité Nacional Peruano de la CIER

En la sesión anterior...

Numpy: Repasemos lo aprendido.

https://docs.google.com/forms/d/e/1FAIpQLSeU2UkkSaj6x8DBTOar61FCQqG9Oid_2p8Y1KXq6Fg1LGwbZA/viewform?usp=header

Introducción a Pandas - DataFrames y Series

PECIER
Comité Nacional Peruano de la CIER

Esta sección introduce al manejo de datos usando **Pandas**, una biblioteca esencial para el sector eléctrico.

Veremos cómo representar lecturas, manipular datos y extraer estadísticas clave para toma de decisiones técnicas.

PECIER
Comité Nacional Peruano de la CIER

Introducción a Pandas - DataFrames y Series

Objetivos

- Comprender la estructura de **Series** y **DataFrames**.
- Realizar **selección, filtrado y operaciones estadísticas**.
- Aplicar Pandas a problemas reales en ingeniería eléctrica.
- Usar este módulo como **material de consulta** durante tus proyectos.

Fundamentos de Pandas

PECIER
Comité Nacional Peruano de la CIER

¿Qué es Pandas?

Pandas es una biblioteca de Python para el análisis y manipulación de datos estructurados (filas y columnas). Útil para tareas como:

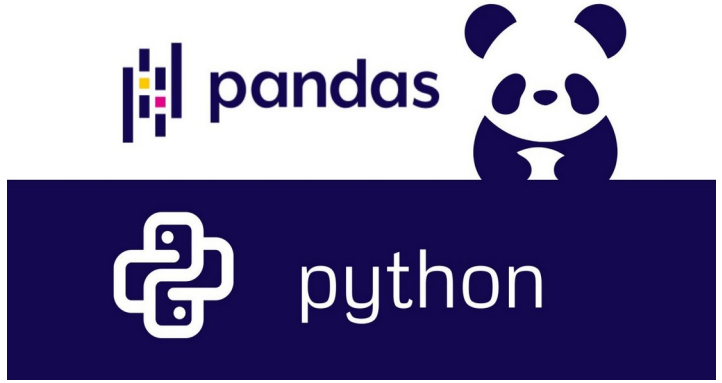
- Analizar mediciones de potencia en transformadores
- Registrar voltajes y corrientes en diferentes fases
- Procesar datos de consumo en instalaciones.

Fundamentos de Pandas

PECIER
Comité Nacional Peruano de la CIER

Instalación (si no está instalada):

```
pip install pandas
```



Recordatorio: pip es el manejador de paquetes (extensiones) de Python. Se usa en la línea de comando.

Fundamentos de Pandas

PECIER
Comité Nacional Peruano de la CIER

Estructuras Básicas

Una **Serie** en Pandas es un arreglo unidimensional con etiquetas (índices). Sirve para representar una sola variable a lo largo del tiempo, como la temperatura o el voltaje.

```
import pandas as pd

mediciones_voltaje = pd.Series([220.5, 221.3, 219.8, 220.2],
                               name='Voltaje (V)',
                               index=['08:00', '09:00', '10:00',
'11:00'])

print(mediciones_voltaje)
```

```
08:00 220.5
09:00 221.3
10:00 219.8
11:00 220.2
Name: Voltaje (V), dtype: float64
```

Fundamentos de Pandas

PECIER
Comité Nacional Peruano de la CIER

DataFrame

Un **DataFrame** es una tabla bidimensional donde cada columna puede tener un tipo distinto de datos (números, cadenas, fechas, etc.). Representa un conjunto de mediciones eléctricas simultáneas.

```
datos_electricos = {
    'Hora': ['08:00', '09:00', '10:00', '11:00'],
    'Voltaje (V)': [220.5, 221.3, 219.8, 220.2],
    'Corriente (A)': [10.2, 9.8, 10.5, 11.1],
    'Potencia (W)': [2249.1, 2168.74, 2307.9, 2444.22]
}

df_mediciones = pd.DataFrame(datos_electricos)
print(df_mediciones)
```

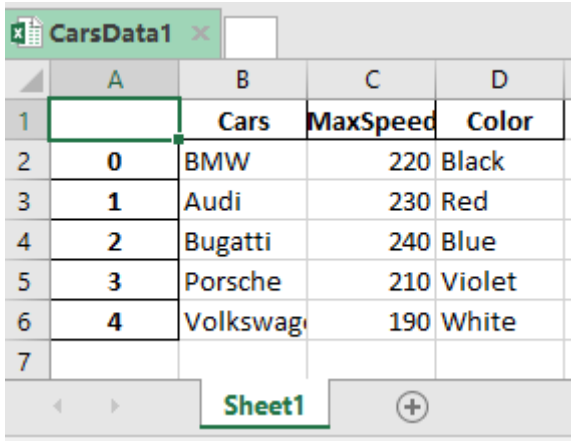
Esta es una de las formas de definir un dataframe

Fundamentos de Pandas

PECIER
Comité Nacional Peruano de la CIER

DataFrame: Analogía

```
datos_electricos = {
    'Hora': ['08:00', '09:00', '10:00',
'11:00'],
    'Voltaje (V)': [220.5, 221.3, 219.8,
220.2],
    'Corriente (A)': [10.2, 9.8, 10.5, 11.1],
  'Potencia (W)': [2249.1, 2168.74, 2307.9,
2444.22]
}

df_mediciones = pd.DataFrame(datos_electricos)
print(df_mediciones)
```



Selección y Filtrado de Datos

PECIER
Comité Nacional Peruano de la CIER

Acceso a Datos con `.loc[]` y `.iloc[]`

En Pandas, existen dos formas fundamentales de seleccionar datos:

Método	Tipo de acceso	¿Usa nombres?	¿Incluye el final del rango?
`.loc[]`	Acceso por **etiquetas**	Sí	Sí
`.iloc[]`	Acceso por **posiciones**	No	No

Selección y Filtrado de Datos

PECIER
Comité Nacional Peruano de la CIER

`.loc[]`: Por etiquetas (nombres de filas y columnas)

Usa etiquetas o nombres del índice o columnas para seleccionar datos. Es más legible.

Selección y Filtrado de Datos

PECIER
Comité Nacional Peruano de la CIER

`.loc[]`: Por etiquetas (nombres de filas y columnas)

Usa etiquetas o nombres del índice o columnas para seleccionar datos. Es más legible.

`.iloc[]`: Por posición (como en listas)

Usa números enteros para acceder a posiciones de filas y columnas. Es útil cuando no sabes los nombres.

Selección y Filtrado de Datos

PECIER
Comité Nacional Peruano de la CIER

Comparación visual

Lo que quieres hacer	`.loc[]`	`.iloc[]`
Fila por nombre	`df.loc['09:00']`	—
Fila por posición	—	`df.iloc[1]`
Rango de filas	`df.loc['09:00':'10:00']`	`df.iloc[1:3]`
Columna por nombre	`df.loc[:, 'Voltaje (V)']`	`df.iloc[:, 0]`
Varias columnas	`df.loc[:, ['Corriente (A)', 'Potencia (W)']]`	`df.iloc[:, 1:3]`
Fila y columna específica	`df.loc['09:00', 'Voltaje (V)']`	`df.iloc[1, 0]`

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia5_parte1

colab.research.google.com

PECIER
Comité Nacional Peruano de la CIER

SESIÓN 5
Parte 2

PECIER
Comité Nacional Peruano de la CIER

Filtrado de Datos

Filtrar significa seleccionar filas que cumplen una condición. Es clave para:

- Identificar sobrecargas
- Detectar caídas de voltaje
- Evaluar horas pico

Selección y Filtrado de Datos

PECIER
Comité Nacional Peruano de la CIER

Filtro simple

```
df_mediciones[df_mediciones['Voltaje (V)'] > 220]
```

Selección y Filtrado de Datos

PECIER
Comité Nacional Peruano de la CIER

Filtro compuesto

Para combinar condiciones se usan los operadores:

- **&** (AND)
- **|** (OR)
- **~** (NOT)

Selección y Filtrado de Datos

PECIER
Comité Nacional Peruano de la CIER

```
condicion = (df_mediciones['Voltaje (V)'] > 219) & (df_mediciones['Corriente (A)']
< 11)
df_mediciones[condicion]
```

PECIER
Comité Nacional Peruano de la CIER

Operaciones Estadísticas

Estadísticas Básicas

Pandas incluye métodos para obtener estadísticas rápidas:

```
df_mediciones.describe()
```

PECIER
Comité Nacional Peruano de la CIER

Operaciones Estadísticas

Incluye:

- count: cantidad de datos
- mean: promedio
- std: desviación estándar
- min, max: valor mínimo y máximo
- 25%, 50%, 75%: percentiles

PECIER
Comité Nacional Peruano de la CIER

Operaciones Estadísticas

Otras Operaciones Útiles

Método	Descripción
`.mean()`	Promedio
`.max()`	Máximo
`.min()`	Mínimo
`.std()`	Desviación estándar
`.sum()`	Suma total
`.median()`	Mediana
`.value_counts()`	Frecuencias (para categóricos)
`.corr()`	Correlación entre columnas numéricas

PECIER
Comité Nacional Peruano de la CIER

Operaciones Estadísticas

```
print(f"Voltaje promedio: {df_mediciones['Voltaje
(V)'].mean():.2f} V") print(f"Potencia máxima:
{df_mediciones['Potencia (W)'].max()} W")
```

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia5_parte2

colab.research.google.com

PECIER
Comité Nacional Peruano de la CIER

SESIÓN 5
Parte 3

PECIER
Comité Nacional Peruano de la CIER

Análisis Temporal y Series de Tiempo

Carga de Datos Reales

```
tensiones = pd.read_csv('tensiones.csv',
                        parse_dates=['fecha'],
index_col='fecha')
print(consumo.head())
```

PECIER
Comité Nacional Peruano de la CIER

Análisis Temporal y Series de Tiempo

Agrupamiento por Intervalos (resample)

Con datos por hora o minuto, puedes agrupar por día ('D'), semana ('W') o mes ('M').

```
tension_diaria =
tensiones['tension1'].resample('D').mean()
```

PECIER
Comité Nacional Peruano de la CIER

Análisis Temporal y Series de Tiempo

Visualización Rápida

Con datos por hora o minuto, puedes agrupar por día ('D'), semana ('W') o mes ('M').

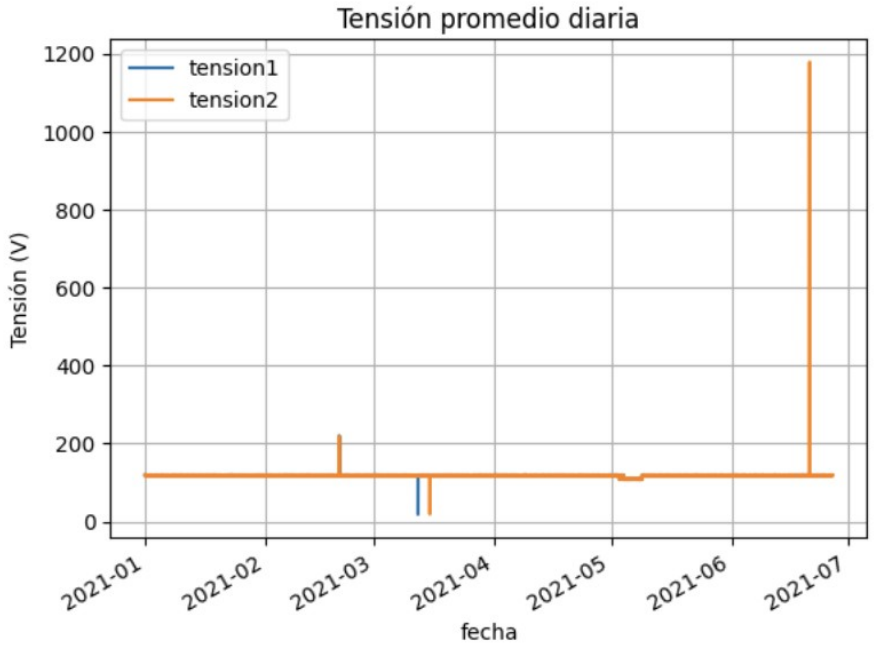
```
import matplotlib.pyplot as plt

consumo_diario.plot(title='Consumo Promedio Diario’)
plt.ylabel('Potencia (W)’)
plt.grid()
plt.show()
```

PECIER
Comité Nacional Peruano de la CIER

Análisis Temporal y Series de Tiempo

Visualización Rápida

Con datos por hora o minuto, puedes agrupar por día ('D'), semana ('W') o mes ('M').



PECIER
Comité Nacional Peruano de la CIER

Detección de Anomalías

Identificar valores anómalos permite detectar problemas como picos, caídas de voltaje o fallas de sensores.

```
media = consumo['Voltaje (V)'].mean() std = consumo['Voltaje (V)'].std()

anomalias = consumo[(consumo['Voltaje (V)'] < media - 3std) | (consumo['Voltaje (V)'] > media
+ 3std)]

print(f"Anomalías detectadas: {len(anomalias)}")
```

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia5_parte3

colab.research.google.com

PECIER
Comité Nacional Peruano de la CIER

SESIÓN 5
Parte 4

PECIER
Comité Nacional Peruano de la CIER

Filtrado, Agrupación y Agregación con Pandas y NumPy

Esta sección tiene como objetivo cubrir los temas de manipulación y transformación de datos eléctricos con Pandas, aplicación de filtros condicionales simples y complejos y manipulación de dataframes.

PECIER
Comité Nacional Peruano de la CIER

Filtrado, Agrupación y Agregación con Pandas y NumPy

Preparación del Entorno

Introducción

En el sector eléctrico es común analizar grandes volúmenes de datos horarios: voltajes, corrientes, potencias, distorsión armónica, etc. Vamos a simular datos de una instalación eléctrica trifásica con mediciones horarias durante 2 días.

PECIER
Comité Nacional Peruano de la CIER

Filtrado, Agrupación y Agregación con Pandas y NumPy

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

np.random.seed(0)  # reproducibilidad

n = 48  # 2 días, datos cada hora
fechas = pd.date_range('2023-01-01', periods=n, freq='H')

data = {
    'Timestamp': fechas,
    'Voltaje (V)': np.random.normal(220, 4, n),
    'Corriente (A)': np.random.normal(10, 1.5, n),
    'Fase': np.random.choice(['R', 'S', 'T'], n),
    'Consumo (kWh)': np.random.uniform(1.0, 5.0, n)
}

df = pd.DataFrame(data).set_index('Timestamp')
df.head()
```

PECIER
Comité Nacional Peruano de la CIER

Manipulación de Columnas

Introducción

Muchas veces necesitamos derivar nueva información a partir de las variables originales. Aquí aprenderemos a:

- Calcular columnas nuevas con fórmulas físicas.
- Calcular promedios móviles y desviaciones estándar.
- Clasificar datos numéricos en niveles.
- Normalizar datos para análisis comparativos.

PECIER
Comité Nacional Peruano de la CIER

Manipulación de Columnas

Potencia Aparente Estimada

Concepto: La potencia aparente en sistemas eléctricos se estima multiplicando voltaje por corriente.

```
df['Potencia (VA)'] = df['Voltaje (V)'] * df['Corriente
(A)']
```

PECIER
Comité Nacional Peruano de la CIER

Manipulación de Columnas

Media Móvil

Concepto: Una media móvil suaviza las fluctuaciones y ayuda a ver tendencias.

```
df['Voltaje_prom_6'] = df['Voltaje
(V)'].rolling(window=6).mean()
```

PECIER
Comité Nacional Peruano de la CIER

Manipulación de Columnas

Desviación Estándar Móvil

Concepto: Mide la variabilidad local de la señal. Aunque vamos a eliminarla luego, antes es buena práctica guardar el DataFrame:

```
df_backup = df.copy()  # recomendación antes de eliminar
columnas
df['Voltaje_std_6'] = df['Voltaje
(V)'].rolling(window=6).std()
df.drop(columns='Voltaje_std_6', inplace=True)
```

PECIER
Comité Nacional Peruano de la CIER

Manipulación de Columnas

Normalización (Z-Score)

Concepto: Transforma la serie de tiempo en una escala centrada en cero, útil para comparación entre variables.

```
df['Voltaje_norm'] = (df['Voltaje (V)'] - df['Voltaje (V)'].mean()) / df['Voltaje (V)'].std()
```

PECIER
Comité Nacional Peruano de la CIER

Manipulación de Columnas

Clasificación Categórica de Voltaje

Concepto: Clasificamos los valores en categorías según su distancia a la media.

```
media = df['Voltaje (V)'].mean()
std = df['Voltaje (V)'].std()

def clasificar_voltaje(v):
    if v < media - std:
        return 'Bajo'
    elif v > media + std:
        return 'Alto'
    else:
        return 'Medio'

df['Nivel Voltaje'] = df['Voltaje (V)'].apply(clasificar_voltaje)
```

PECIER
Comité Nacional Peruano de la CIER

Manipulación de Columnas

Clasificación con NumPy: `np.where()`

¿Qué es np.where()? Es una función de NumPy que permite crear columnas condicionales de forma rápida.

Ejemplo: Indicador de voltaje seguro (seguro si es menor a 230 V):

```
df['Voltaje Seguro'] = np.where(df['Voltaje (V)'] < 230, 'Sí',
'No')
```

Esto es equivalente a:

```
# Sin np.where (más largo)
df['Voltaje Seguro'] = df['Voltaje (V)'].apply(lambda v: 'Sí' if v < 230
else 'No')
```

PECIER
Comité Nacional Peruano de la CIER

Fusión de DataFrames

Introducción

Muchas veces tenemos datos complementarios en distintos DataFrames. La unión de DataFrames permite integrarlos.

PECIER
Comité Nacional Peruano de la CIER

Fusión de DataFrames

Unión por Índice (.join())

```
thd = pd.DataFrame({
    'Timestamp': fechas,
    'THD': np.random.uniform(3, 7, n)
}).set_index('Timestamp')

df = df.join(thd)
```

PECIER
Comité Nacional Peruano de la CIER

Fusión de DataFrames

Unión por Índice (.join())

Para que funcione:

- Ambos DataFrames deben tener el mismo índice (DatetimeIndex).
- Si hay diferencias en los índices, el resultado puede contener NaN.

PECIER
Comité Nacional Peruano de la CIER

Fusión de DataFrames

Unión con `merge()`

```
mediciones_extra = pd.DataFrame({
    'Timestamp': fechas,
    'Factor_Potencia': np.random.uniform(0.85, 1.0, n)
})

df = df.reset_index().merge(mediciones_extra,
on='Timestamp').set_index('Timestamp')
```

Para que funcione:

- Las columnas clave deben tener el mismo nombre y tipo.
- Puede usarse how='left', how='right' o how='outer' según la necesidad.

PECIER
Comité Nacional Peruano de la CIER

Fusión de DataFrames

Extracción Parcial de información de un Dataframe

```
df_partes = df[['Voltaje (V)', 'Corriente (A)', 'Fase']]
df_rango = df['2023-01-01 08:00':'2023-01-01 20:00']
```

PECIER
Comité Nacional Peruano de la CIER

Filtrado de Datos

Introducción

Filtrar permite trabajar solo con los datos que cumplen ciertas condiciones.

```
df[df['Corriente (A)'] > 11]

df[(df['Fase'] == 'S') & (df['Consumo (kWh)'].between(3, 4))]

df['2023-01-01']
```

PECIER
Comité Nacional Peruano de la CIER

Filtrado de Datos

In [17]:
```
df[df['Corriente (A)'] > 11]
```

Out[17]:

Timestamp	Voltaje (V)	Corriente (A)	Fase	Consumo (kWh)	Potencia (VA)	Voltaje_prom_6	Voltaje_norm	Nivel Voltaje	Voltaje Seguro	THD	Factor_Potencia
2023-01-01 22:00:00	223.457745	11.093636	S	4.712325	2478.958847	218.812586	0.601587	Medio	Sí	6.811167	0.994236
2023-01-02 00:00:00	229.079018	11.709101	R	1.127356	2682.309371	219.759039	1.844918	Alto	Sí	3.862031	0.992398
2023-01-02 09:00:00	212.076814	11.351240	T	3.920488	2407.334758	220.444444	-1.915682	Bajo	Sí	4.496680	0.902085
2023-01-02 12:00:00	224.921163	12.232378	R	1.839375	2751.320748	219.032205	0.925271	Medio	Sí	5.347137	0.921756
2023-01-02 13:00:00	224.809519	12.843834	R	1.744772	2887.416096	219.581683	0.900577	Medio	Sí	6.455422	0.924609
2023-01-02 14:00:00	218.450693	11.768169	R	4.777490	2570.764748	219.915323	-0.505888	Medio	Sí	3.470127	0.945921
2023-01-02 17:00:00	214.319928	11.581678	S	1.909659	2482.184310	219.516313	-1.419543	Bajo	Sí	5.867439	0.973318
2023-01-02 19:00:00	227.803102	11.833668	R	1.232117	2695.746184	218.057536	1.562707	Alto	Sí	5.261685	0.926698
2023-01-02 21:00:00	218.247703	11.464959	T	2.247184	2502.200867	217.885472	-0.550786	Medio	Sí	3.579391	0.864677
2023-01-02 23:00:00	223.109961	11.059860	S	2.511007	2467.564883	219.214316	0.524664	Medio	Sí	4.422451	0.995938

PECIER
Comité Nacional Peruano de la CIER

Filtrado de Datos

In [18]:
```
df[(df['Fase'] == 'S') & (df['Consumo (kWh)'].between(3, 4))]
```

Out[18]:

	Voltaje (V)	Corriente (A)	Fase	Consumo (kWh)	Potencia (VA)	Voltaje_prom_6	Voltaje_norm	Nivel Voltaje	Voltaje Seguro	THD	Factor_Potencia
Timestamp											
2023-01-01 05:00:00	216.090888	8.229052	S	3.789715	1778.223098	224.182747	-1.027837	Bajo	Sí	5.146317	0.932829
2023-01-01 15:00:00	221.334697	7.410576	S	3.084146	1640.217617	222.172378	0.132005	Medio	Sí	5.353268	0.892260
2023-01-02 02:00:00	220.183034	10.603512	S	3.485914	2334.713546	221.091358	-0.122723	Medio	Sí	5.923423	0.969880
2023-01-02 22:00:00	214.988819	10.534550	S	3.785374	2264.810372	217.749310	-1.271596	Bajo	Sí	4.952225	0.979329

PECIER
Comité Nacional Peruano de la CIER

Agrupación y Agregación

Introducción

Agrupar permite calcular estadísticas por categorías (como por fase o por hora).

```
df.groupby('Fase')[['Voltaje (V)', 'Corriente (A)']].mean()

df.groupby('Fase').agg({
    'Voltaje (V)': ['mean', 'max'],
    'Corriente (A)': ['std'],
    'Consumo (kWh)': ['sum']
})
```

PECIER
Comité Nacional Peruano de la CIER

Agrupación y Agregación

Introducción

Agrupar permite calcular estadísticas por categorías (como por fase o por hora).

In [23]:
```
df.groupby('Fase').agg({
    'Voltaje (V)': ['mean', 'max'],
    'Corriente (A)': ['std'],
    'Consumo (kWh)': ['sum']
})
```

Out[23]:

	Voltaje (V)		Corriente (A)	Consumo (kWh)
	mean	max	std	sum
Fase				
R	221.772391	229.079018	1.528422	44.848753
S	220.379685	228.963573	1.205664	50.936867
T	220.016130	226.131117	1.190419	44.261812

PECIER
Comité Nacional Peruano de la CIER

Agrupación y Agregación

Por hora:

```
df['Hora'] = df.index.hour
df.groupby('Hora')['Consumo
(kWh)'].mean()
```

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia5_parte4

colab.research.google.com

PECIER
Comité Nacional Peruano de la CIER

SESIÓN 5
Parte 5

PECIER
Comité Nacional Peruano de la CIER

Visualización de Datos con Matplotlib y Seaborn para aplicaciones del Sector Eléctrico

Objetivos

- Dominar los fundamentos de visualización con Matplotlib
- Comprender las ventajas de Seaborn en el análisis estadístico
- Crear visualizaciones profesionales para análisis eléctricos

PECIER
Comité Nacional Peruano de la CIER

Introducción

¿Por qué es importante visualizar datos?

Los sistemas eléctricos generan grandes volúmenes de datos, como:

- Mediciones de voltaje, corriente, frecuencia y potencia
- Calidad de energía y distorsiones armónicas
- Cargas por fases, balanceo de redes, entre otros

PECIER
Comité Nacional Peruano de la CIER

Introducción

Una buena visualización permite:

- Identificar picos, caídas o patrones repetitivos
- Detectar fallos o condiciones anómalas
- Tomar decisiones rápidas y comunicar hallazgos

PECIER
Comité Nacional Peruano de la CIER

Bibliotecas Matplotlib y Seaborn

¿Qué es Matplotlib?

Matplotlib es la biblioteca más antigua y flexible de gráficos en Python. Permite crear gráficos altamente personalizados como líneas, barras, histogramas, dispersión, etc.

- Módulo principal: pyplot
- Pensado como una herramienta estilo MATLAB
- Alta personalización manual

PECIER
Comité Nacional Peruano de la CIER

Bibliotecas Matplotlib y Seaborn

¿Qué es Seaborn?

- Seaborn está construida sobre Matplotlib, y facilita la creación de gráficos estadísticos. Se destaca por:
- Gráficos estéticos por defecto
- Integración directa con pandas
- Funcionalidades estadísticas como regresiones, boxplots y mapas de calor

PECIER
Comité Nacional Peruano de la CIER

Bibliotecas Matplotlib y Seaborn

¿Qué es Seaborn?

Característica	Matplotlib	Seaborn
Control detallado	Sí	Parcial
Fácil de usar	Parcial	Sí
Gráficos estadísticos	No	Sí
Integración con Pandas	Sí	Sí

Recomendación: Usar Matplotlib para control completos. Usar Seaborn para análisis rápidos y exploración.

PECIER
Comité Nacional Peruano de la CIER

Configuración Inicial

Cargar bibliotecas y crear dataset de ejemplo

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

plt.style.use('seaborn-v0_8')  # Estilo
predeterminado moderno
sns.set_palette("husl")        # Paleta de
colores para Seaborn
%matplotlib inline
```

PECIER
Comité Nacional Peruano de la CIER

Configuración Inicial

Crear datos de simulación

```
np.random.seed(42)

horas = pd.date_range('2023-06-01', periods=24, freq='H')
datos = {
    'Hora': horas,
    'Voltaje': np.random.normal(220, 5, 24),
    'Corriente': np.random.gamma(5, 0.5, 24),
    'Potencia': np.random.uniform(1000, 5000, 24),
    'Fase': np.random.choice(['R', 'S', 'T'], 24)
}
df = pd.DataFrame(datos).set_index('Hora')
```

PECIER
Comité Nacional Peruano de la CIER

Visualización con Matplotlib

Gráficos de Líneas

¿Cuándo son útiles estos gráficos?

- Series temporales como voltaje diario, temperatura, potencia
- Comparación con valores nominales

PECIER
Comité Nacional Peruano de la CIER

Visualización con Matplotlib

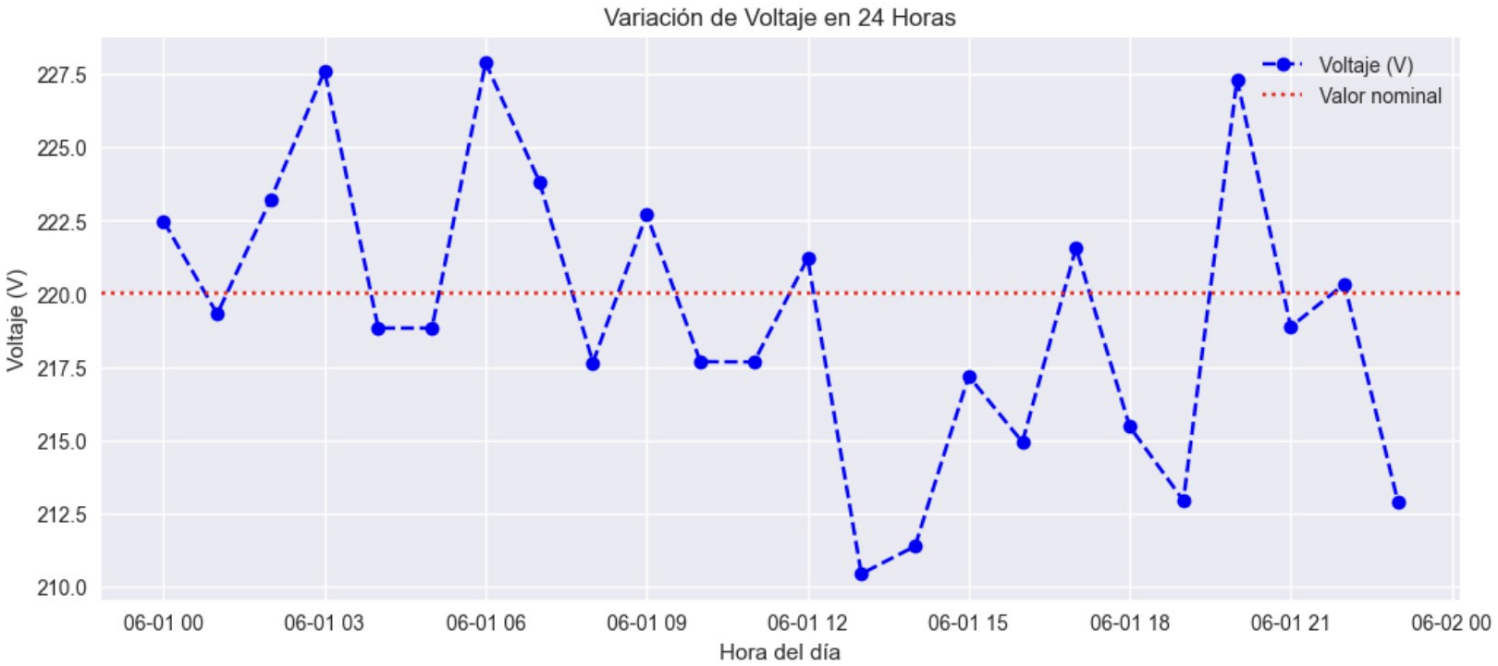
Ejemplo: Voltaje a lo largo del día

```
plt.figure(figsize=(12, 5))
plt.plot(df.index, df['Voltaje'],
         marker='o',              # Estilo del punto (otros: 's' cuadrado, '^' tria
         linestyle='--',          # '--' = línea punteada. También: '-' sólida, '-
         color='b',               # 'b' = azul. Otros: 'r' rojo, 'g' verde, 'k' neg
         label='Voltaje (V)')     # Etiqueta de leyenda

plt.axhline(220, color='r', linestyle=':', label='Valor nominal')
plt.title('Variación de Voltaje en 24 Horas')
plt.xlabel('Hora del día')
plt.ylabel('Voltaje (V)')
plt.legend()
plt.grid(True)
plt.show()
```

PECIER
Comité Nacional Peruano de la CIER

Visualización con Matplotlib

Ejemplo: Voltaje a lo largo del día



Visualización con Matplotlib

PECIER
Comité Nacional Peruano de la CIER

Explicación de cada línea

Elemento	Significado
`plt.plot()`	Crea gráfico de líneas
`marker='o'`	Marca cada punto con círculo
`linestyle='--'`	Línea punteada
`color='b'`	Color azul
`axhline()`	Línea horizontal (valor de referencia)

Visualización con Matplotlib

PECIER
Comité Nacional Peruano de la CIER

Histogramas

¿Cuándo son útiles los histogramas?

- Para visualizar distribución de una variable (frecuencia)
- Útil para corriente, THD, factor de potencia

Visualización con Matplotlib

PECIER
Comité Nacional Peruano de la CIER

Histogramas

Ejemplo: Distribución de corriente

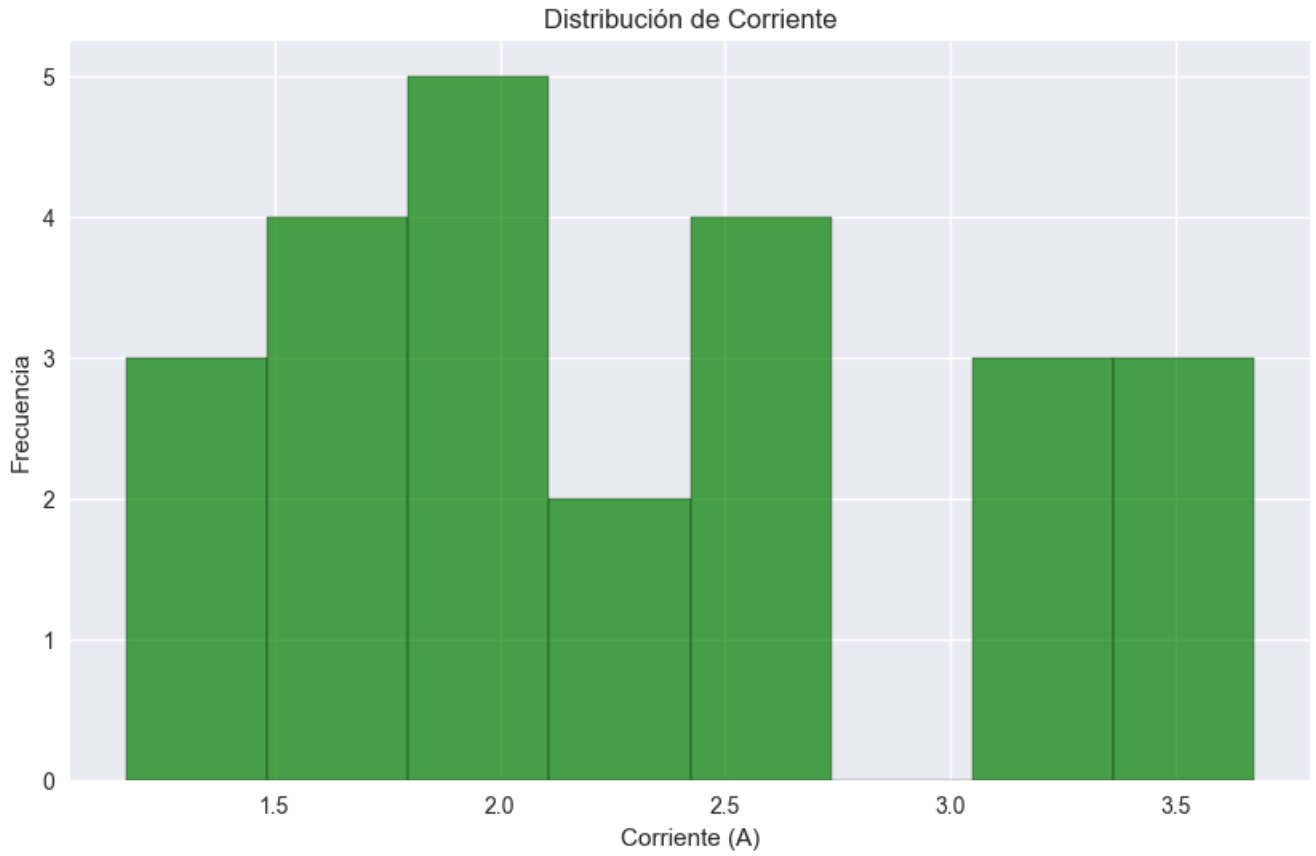
```
plt.figure(figsize=(10, 6))
plt.hist(df['Corriente'],
        bins=8,              # Número de bloques
        color='green',
        alpha=0.7,           # Transparencia (0 = invisible, 1 = opaco)
        edgecolor='black')    # Borde de las barras
plt.title('Distribución de Corriente')
plt.xlabel('Corriente (A)')
plt.ylabel('Frecuencia')
plt.show()
```

PECIER
Comité Nacional Peruano de la CIER

Visualización con Matplotlib

Histogramas

Ejemplo: Distribución de corriente



Visualización con Matplotlib

PECIER
Comité Nacional Peruano de la CIER

Gráficos de Dispersión + Regresión

¿Cuándo son útiles?

- Para analizar correlación entre dos variables
- Ej: corriente vs potencia, THD vs temperatura

Visualización con Matplotlib

PECIER
Comité Nacional Peruano de la CIER

Gráficos de Dispersión + Regresión

Ejemplo:

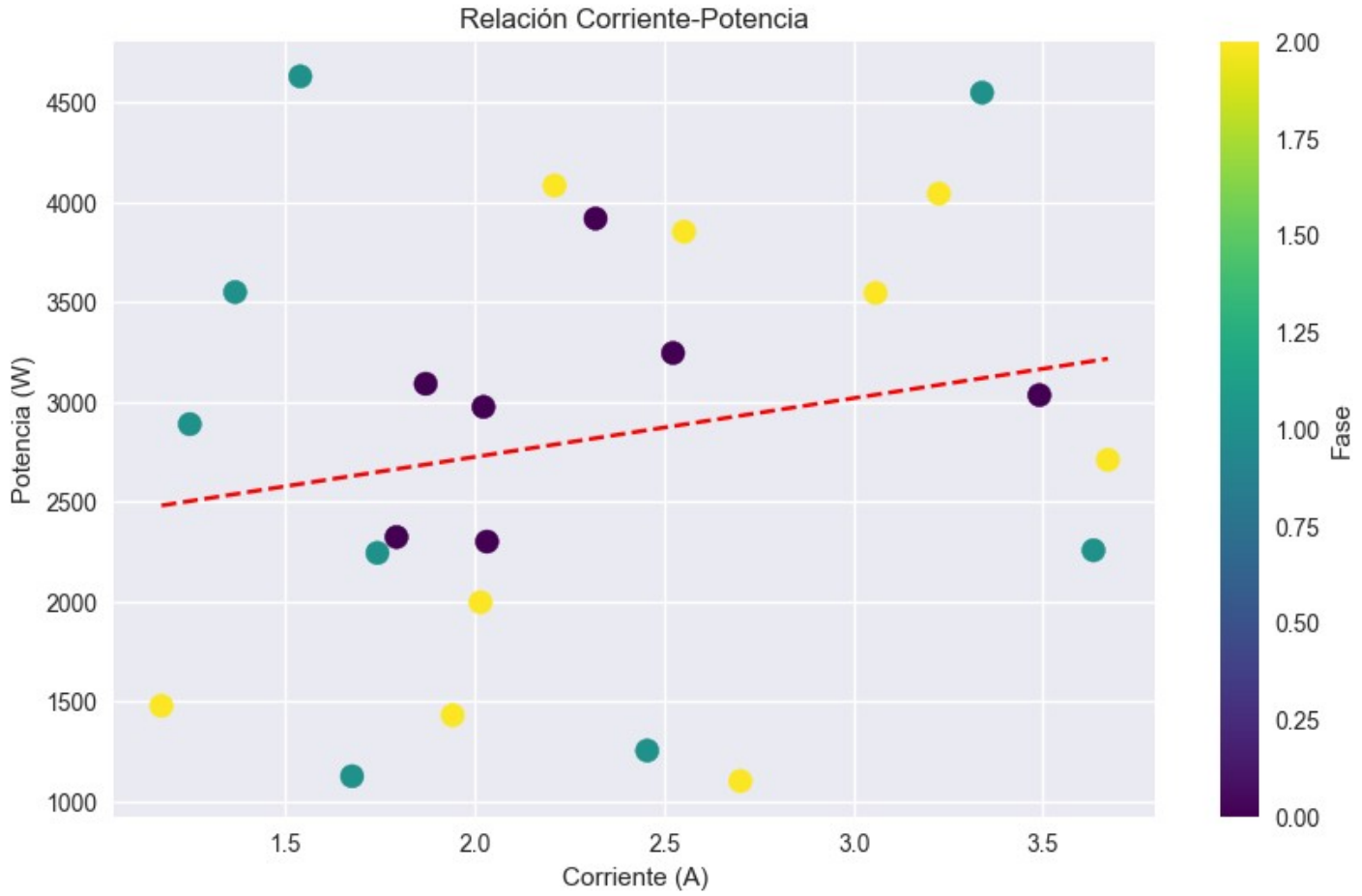
```
plt.figure(figsize=(10, 6))
plt.scatter(df['Corriente'], df['Potencia'],
            c=pd.factorize(df['Fase'])[0], # Codifica R/S/T como
0/1/2
            cmap='viridis',                # Mapa de color
            s=100)                         # Tamaño de puntos
plt.colorbar(label='Fase')
plt.plot(np.unique(df['Corriente']),
         np.poly1d(np.polyfit(df['Corriente'], df['Potencia'], 1))
(np.unique(df['Corriente'])),
         'r--')  # Línea de regresión (roja punteada)
plt.title('Relación Corriente-Potencia')
plt.xlabel('Corriente (A)')
plt.ylabel('Potencia (W)')
plt.show()
```

PECIER
Comité Nacional Peruano de la CIER

Visualización con Matplotlib

Gráficos de Dispersión + Regresión

Ejemplo:



Visualización con Seaborn

PECIER
Comité Nacional Peruano de la CIER

Boxplots + Swarmplots

¿Cuándo usarlo?

- Comparar valores por grupo (fases, días, zonas)
- Ver valores extremos (outliers) y mediana

Visualización con Seaborn

PECIER
Comité Nacional Peruano de la CIER

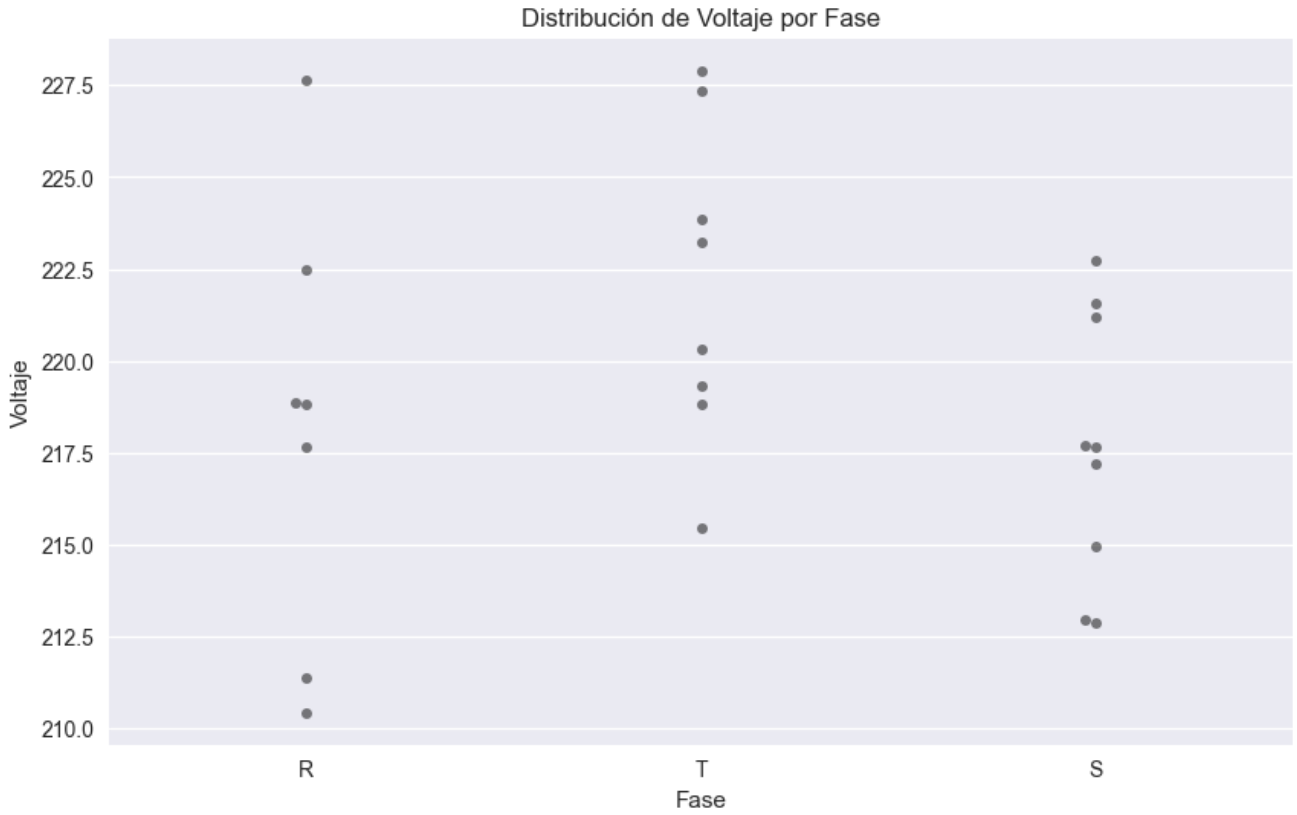
Ejemplo:

```
plt.figure(figsize=(10, 6))
#sns.boxplot(x='Fase', y='Voltaje', data=df.reset_index(),
#            width=0.3, notch=True)  # notch=True muestra intervalo de
confianza
sns.swarmplot(x='Fase', y='Voltaje', data=df.reset_index(),
              color='black', alpha=0.5)  # Puntos individuales
plt.title('Distribución de Voltaje por Fase')
plt.show()
```

Visualización con Seaborn

PECIER
Comité Nacional Peruano de la CIER

Ejemplo:



Recomendación: Ver https://www.geeksforgeeks.org/python/swarmplot-using-seaborn-in-python/

Visualización con Seaborn

PECIER
Comité Nacional Peruano de la CIER

Mapa de calor de correlaciones

¿Cuándo usar?

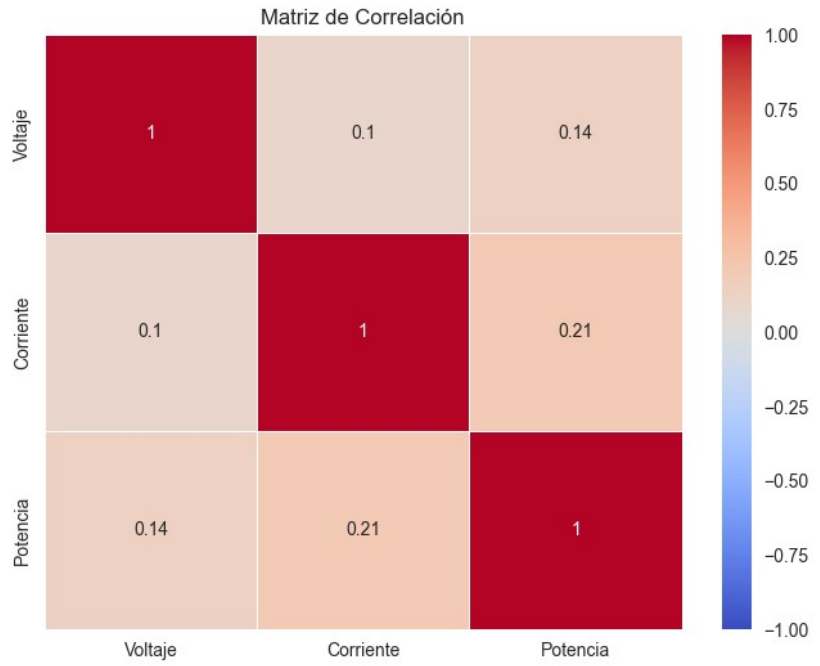
- Visualizar qué variables están correlacionadas
- Identificar relaciones lineales

PECIER
Comité Nacional Peruano de la CIER

Visualización con Seaborn

Ejemplo:

```
corr = df[['Voltaje', 'Corriente', 'Potencia']].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(corr, annot=True, cmap='coolwarm', vmin=-1, vmax=1, linewidths=0.5
plt.title('Matriz de Correlación')
plt.show()
```



Visualización Temporal Avanzada

PECIER
Comité Nacional Peruano de la CIER

Ejemplo con datos de una semana

Este gráfico muestra cómo varía el voltaje cada día, comparando por hora.

```
semana = pd.date_range('2023-06-01', periods=7*24, freq='H')
df_semana = pd.DataFrame({
    'Timestamp': semana,
    'Voltaje': np.random.normal(220, 3, len(semana)) +
5*np.sin(2*np.pi*semana.hour/24),
    'Consumo': np.random.gamma(5, 100, len(semana)),
    'Dia': semana.day_name(),
    'Hora': semana.hour
}).set_index('Timestamp')
```

Visualización Temporal Avanzada

PECIER
Comité Nacional Peruano de la CIER

Ejemplo con datos de una semana

Este gráfico muestra cómo varía el voltaje cada día, comparando por hora.

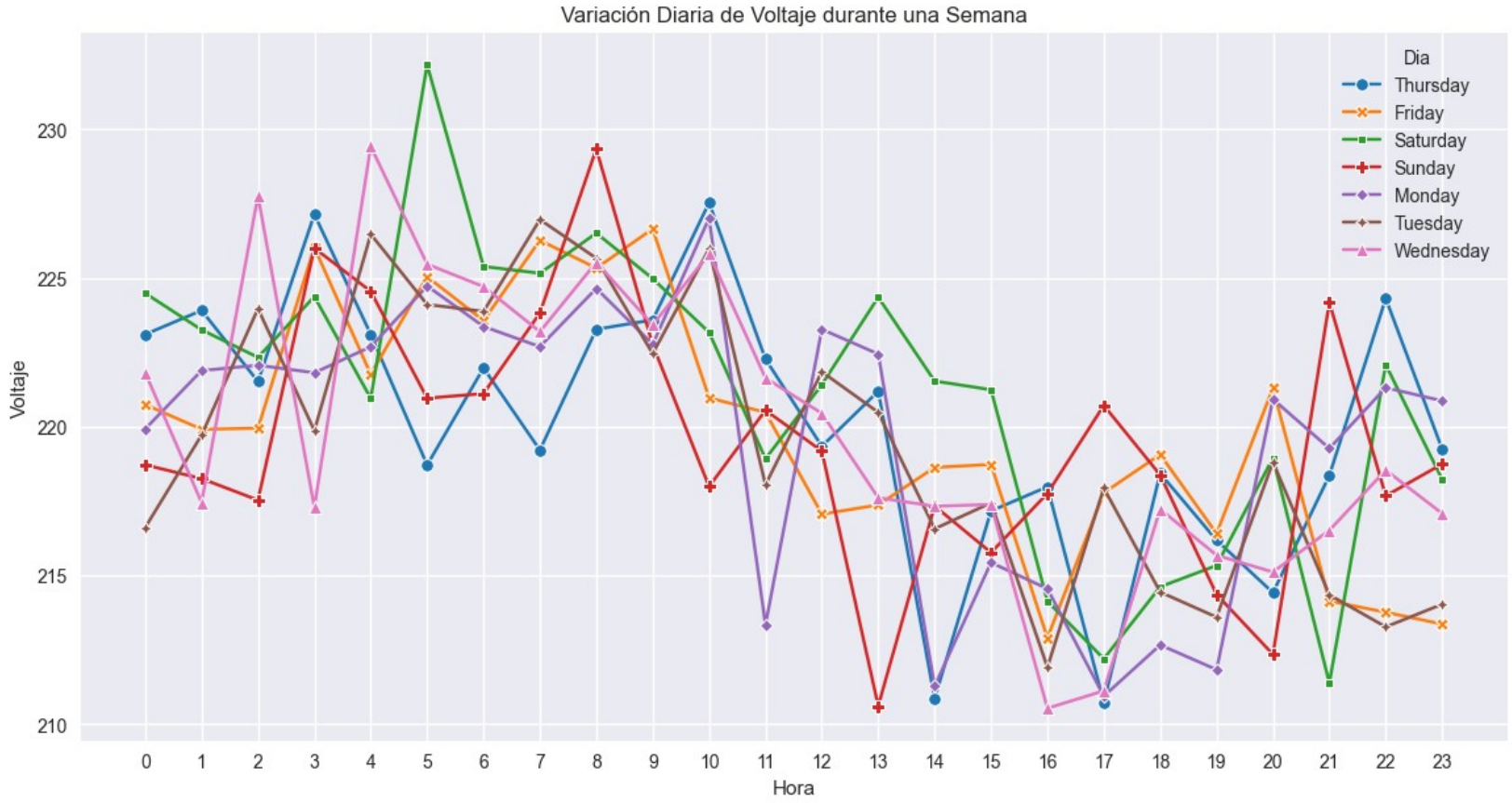
```
plt.figure(figsize=(14, 7))
sns.lineplot(data=df_semana.reset_index(),
             x='Hora',
             y='Voltaje',
             hue='Dia',
             style='Dia',
             markers=True,
             dashes=False,
             palette='tab10')
plt.title('Variación Diaria de Voltaje durante una Semana')
plt.xticks(range(0, 24))
plt.grid(True)
plt.show()
```

PECIER
Comité Nacional Peruano de la CIER

Visualización Temporal Avanzada

Ejemplo con datos de una semana

Este gráfico muestra cómo varía el voltaje cada día, comparando por hora.



Dashboard de Monitoreo Eléctrico

PECIER
Comité Nacional Peruano de la CIER

¿Qué es un dashboard?

Un **dashboard** (tablero de control) es una visualización compuesta que presenta múltiples gráficos para monitorear variables clave. En el contexto eléctrico, permite observar:

- Tendencias generales
- Anomalías en tiempo real
- Distribuciones y correlaciones entre variables

Dashboard de Monitoreo Eléctrico

PECIER
Comité Nacional Peruano de la CIER

Ejemplo de un Dashboard

Vamos a construir un dashboard que incluya:

- Serie temporal
- Histograma
- Boxplot
- Mapa de calor
- Dispersión con colores por día

Dashboard de Monitoreo Eléctrico

PECIER
Comité Nacional Peruano de la CIER

Explicación

- Se usa gridspec para organizar el layout (3 filas × 2 columnas)
- Cada gráfico ocupa su propia celda o combinación de celdas
- Las visualizaciones permiten inspección cruzada de variable

PECIER
Comité Nacional Peruano de la CIER

Análisis de Anomalías

¿Qué es una anomalía?

En sistemas eléctricos, una anomalía puede ser:

- Voltaje fuera de rango
- Consumo anormalmente alto o bajo
- Eventos que exceden límites estadísticos

PECIER
Comité Nacional Peruano de la CIER

Análisis de Anomalías

Detección con 2 desviaciones estándar

```
def detectar_anomalias(df):
    mean_v = df['Voltaje'].mean()
    std_v = df['Voltaje'].std()

    df['Anomalia'] = (df['Voltaje'] < mean_v - 2 * std_v) | \
                     (df['Voltaje'] > mean_v + 2 * std_v)

    plt.figure(figsize=(14, 6))
    sns.scatterplot(data=df.reset_index(),
                    x='Timestamp',
                    y='Voltaje',
                    hue='Anomalia',
                    palette={True: 'red', False: 'blue'},
                    style='Anomalia',
                    s=100)
```

Análisis de Anomalías

PECIER
Comité Nacional Peruano de la CIER

Detección con 2 desviaciones estándar

```
plt.axhline(mean_v, color='green', linestyle='--', label='Media')
    plt.axhline(mean_v + 2 * std_v, color='orange', linestyle=':',
label='Límites')
    plt.axhline(mean_v - 2 * std_v, color='orange', linestyle=':')
    plt.title('Detección de Anomalías en Voltaje')
    plt.legend()
    plt.show()

    return df[df['Anomalia']]

anomalias = detectar_anomalias(df_semana)
print(f"Anomalías detectadas: {len(anomalias)}")
```

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia5_parte5

colab.research.google.com

PECIER

Comité Nacional Peruano de la CIER

E-mail: pecier@cier.org