

##Taller N°1 Análisis Datos Python Arturo Vargas Javier Cruzat

```
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

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

fd_001 =
pd.read_csv("/content/drive/MyDrive/CMAPSS/CMAPSS/train_FD001.txt", sep
=" ", header=None)

fd_002 =
pd.read_csv("/content/drive/MyDrive/CMAPSS/CMAPSS/train_FD002.txt", sep
=" ", header=None)

fd_003 =
pd.read_csv("/content/drive/MyDrive/CMAPSS/CMAPSS/train_FD003.txt", sep
=" ", header=None)

fd_004 =
pd.read_csv("/content/drive/MyDrive/CMAPSS/CMAPSS/train_FD004.txt", sep
=" ", header=None)

from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

Experimento 1

El documento explica un método para simular cómo se acumula el daño en los motores de los aviones. Utiliza un modelo de computadora del motor para predecir cómo cambian las lecturas de los sensores a medida que el motor se desgasta. Este desgaste se simula como una disminución en el "flujo" y la "eficiencia" del motor a lo largo del tiempo.

La simulación continúa hasta que el motor alcanza un punto de "falla" predefinido. Los datos generados por esta simulación se utilizaron en una competencia para probar qué tan bien los programas de computadora pueden predecir cuándo fallarán los motores.

Experimento 2

El modelo de Coffin-Manson predice la falla de componentes por fatiga en motores de avión. Considera cómo los cambios de temperatura y la frecuencia de esos cambios afectan la formación de grietas y, finalmente, la falla del componente.

El modelo de Coffin-Manson predice la falla por fatiga considerando:

Ciclos hasta la falla: Cuántos ciclos de tensión soporta el componente.

Frecuencia de ciclado: Rapidez de los ciclos de tensión.

Rango de temperatura: Variación de temperatura en cada ciclo.

Estas variables estiman la resistencia del material a la fatiga.

##Experimento 3 La experiencia FD003 del modelo C-MAPSS simula el funcionamiento de 100 motores turbofan que operan bajo una única condición operativa constante, pero a diferencia de FD001, incorpora múltiples modos de falla al mismo tiempo. Esto significa que distintos componentes del motor (como compresores y turbinas) pueden degradarse simultáneamente.

El desgaste se modela con funciones exponenciales, y se añade ruido realista (de proceso, medición y ensamblaje) para imitar condiciones reales. Cada registro incluye el ciclo de vuelo, el ID del motor, condiciones de operación (fijas) y 21 sensores.

Este conjunto es útil para estudiar cómo interactúan distintas fallas en un entorno estable, ideal para probar modelos de pronóstico que analicen trayectorias de deterioro más complejas sin la interferencia de cambios externos.

El Experimento FD004

simula el daño en motores de turbofan para desarrollar métodos de mantenimiento predictivo. Se simulan 249 motores con condiciones operativas variables (altitud, número Mach, ángulo del acelerador) para crear un escenario realista. El daño se modela como una pérdida de eficiencia y flujo, con factores añadidos para imitar datos reales (ruido, variaciones de ensamblaje, etc.). El dataset resultante incluye lecturas de sensores y condiciones operativas por cada ciclo de vuelo, sin proporcionar directamente la vida útil restante (RUL), la cual debe ser estimada.

Las variables

El dataset FD004 simula motores con:

ID de motor: Identifica cada motor.

Número de ciclo: Indica el uso del motor.

Condiciones operativas: (altitud, ángulo del acelerador) Varían durante la simulación.

Variables de sensor: 21 lecturas simuladas del motor.

Analisis Experimento 1

```
fd_001
{"type": "dataframe", "variable_name": "fd_001"}
fd_001.drop(columns=[26,27], inplace=True)
columns =
['unit_number', 'time_in_cycles', 'setting_1', 'setting_2', 'TRA', 'T2', 'T2
```

```

4', 'T30', 'T50', 'P2', 'P15', 'P30', 'Nf',
'Nc', 'epr', 'Ps30', 'phi', 'NRf', 'NRc', 'BPR', 'farB', 'htBleed', 'Nf_dmd', 'P
CNfR_dmd', 'W31', 'W32']

fd_001.columns = columns

fd_001

{"type": "dataframe", "variable_name": "fd_001"}

def prepare_train_data(data, factor = 0):
    # Hacer una copia del DataFrame original para evitar cambios en el
    DataFrame original
    df = data.copy()

    # Agrupar por 'unit_number' y encontrar el máximo 'time_in_cycles'
    para cada grupo
    # reset_index() se usa para convertir 'unit_number' de un índice a
    una columna
    fd_RUL = df.groupby('unit_number')
    ['time_in_cycles'].max().reset_index()

    # Cambiar el nombre de la columna 'time_in_cycles' a 'max'
    fd_RUL.rename(columns = {'time_in_cycles': 'max'}, inplace = True)

    # Combinar el DataFrame original con fd_RUL basado en
    'unit_number'
    # Esto agrega la columna 'max' al DataFrame original
    df = df.merge(fd_RUL, on=['unit_number'], how='left')

    # Calcular la columna 'RUL' como la diferencia entre 'max' y
    'time_in_cycles'
    df['RUL'] = df['max'] - df['time_in_cycles']

    # Eliminar la columna 'max' ya que ya no es necesaria
    df.drop(columns=['max'], inplace = True)

    # Devolver solo las filas donde 'time_in_cycles' es mayor que el
    factor dado
    return df[df['time_in_cycles'] > factor]

df = prepare_train_data(fd_001)

df

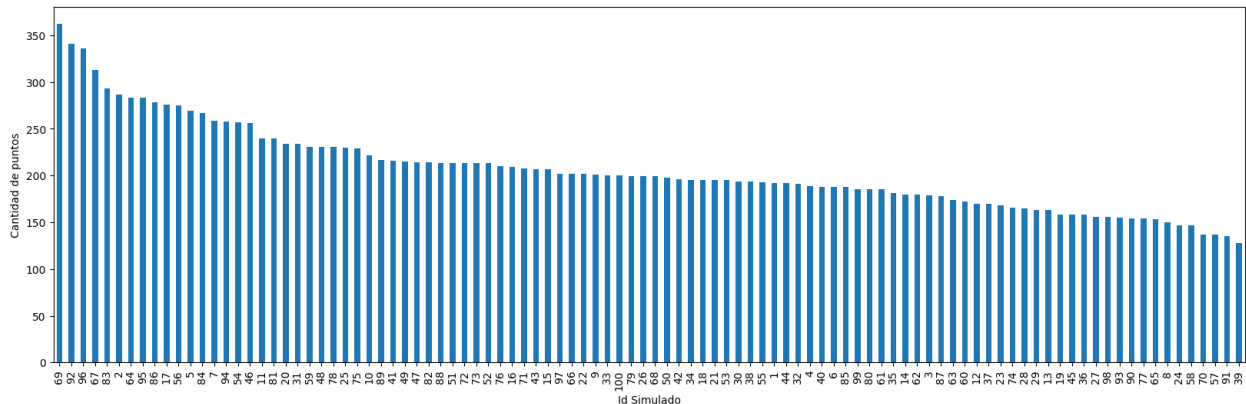
{"type": "dataframe", "variable_name": "df"}

#Veamos cada campaña
plt.figure(figsize=(20,6))
df.unit_number.value_counts().plot.bar()

```

```
plt.xlabel('Id Simulado')
plt.ylabel('Cantidad de puntos')

Text(0, 0.5, 'Cantidad de puntos')
```



##análisis gráfico del conjunto FD001, se observa que sensores como T50 (temperatura de salida de la turbina) presentan una tendencia creciente a medida que aumentan los ciclos, lo que indica un mayor esfuerzo térmico conforme el motor se degrada. Por otro lado, variables como P30 (presión del compresor de alta presión) tienden a disminuir progresivamente, reflejando una pérdida de rendimiento del sistema. Sensores como Nc y Nf muestran leves fluctuaciones o descensos hacia el final del ciclo de vida. Estas tendencias permiten identificar patrones claros de degradación en un entorno sin variabilidad externa, lo que hace de FD001 un caso ideal para estudiar el deterioro de motores con un solo modo de falla.

Analisis experimento 2

```
fd_002 =
pd.read_csv("/content/drive/MyDrive/CMAPSS/CMAPSS/train_FD002.txt", sep
=" ", header=None)

fd_002

{"type": "dataframe", "variable_name": "fd_002"}

fd_002.drop(columns=[26,27], inplace=True)

columns =
['unit_number', 'time_in_cycles', 'setting_1', 'setting_2', 'TRA', 'T2', 'T2
4', 'T30', 'T50', 'P2', 'P15', 'P30', 'Nf',

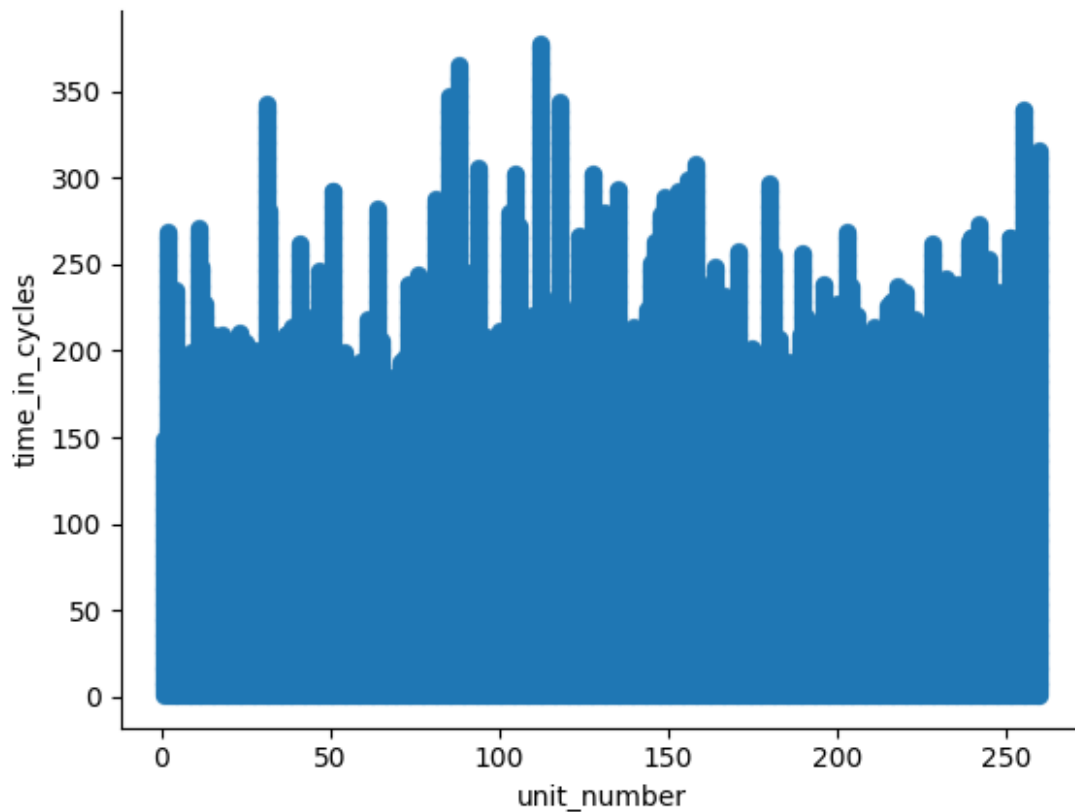
'Nc', 'epr', 'Ps30', 'phi', 'NRf', 'NRc', 'BPR', 'farB', 'htBleed', 'Nf_dmd', 'P
CNfR_dmd', 'W31', 'W32']

fd_002.columns = columns

fd_002

{"type": "dataframe", "variable_name": "fd_002"}
```

```
from matplotlib import pyplot as plt
fd_002.plot(kind='scatter', x='unit_number', y='time_in_cycles', s=32,
alpha=.8)
plt.gca().spines[['top', 'right']].set_visible(False)
```



```
def prepare_train_data(data, factor = 0):
    # Hacer una copia del DataFrame original para evitar cambios en el
    # DataFrame original
    df = data.copy()
    #Agrupar por
```

##Análisis del gráfico En FD002, los gráficos de sensores muestran un comportamiento más disperso en comparación con FD001 debido a las seis condiciones operativas diferentes. Por ejemplo, T50 aún muestra una tendencia creciente general, pero con más variaciones entre motores por las diferencias de entorno. La P30 también disminuye con el tiempo, pero su curva no es tan suave ni uniforme. Se observan saltos o cambios bruscos en sensores como Nc y TRA, que responden a los cambios de altitud y empuje. Esto complica el análisis directo de degradación, ya que es necesario aislar los efectos del entorno para detectar el desgaste real.

##Análisis experimento 3

```
columns =
['unit_number', 'time_in_cycles', 'setting_1', 'setting_2', 'TRA', 'T2', 'T2
```

```

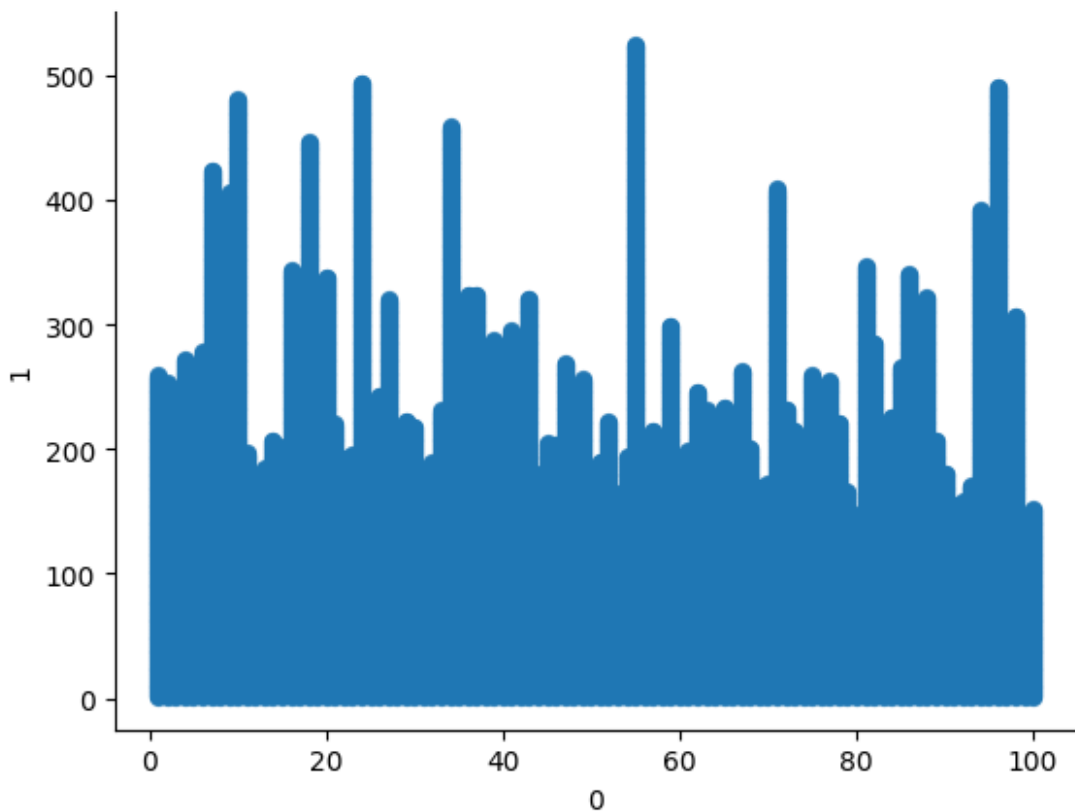
4', 'T30', 'T50', 'P2', 'P15', 'P30', 'Nf',
'Nc', 'epr', 'Ps30', 'phi', 'NRf', 'NRc', 'BPR', 'farB', 'htBleed', 'Nf_dmd', 'P
CNfR_dmd', 'W31', 'W32']

fd_003

{"type": "dataframe", "variable_name": "fd_003"}

from matplotlib import pyplot as plt
fd_003.plot(kind='scatter', x=0, y=1, s=32, alpha=.8)
plt.gca().spines[['top', 'right']].set_visible(False)

```



##Análisis del gráfico

El gráfico muestra la variabilidad del RUL inicial de 100 motores en el dataset FD003, con valores entre 0 y más de 500 ciclos. Esta dispersión indica diferencias en la durabilidad de los motores y la necesidad de un buen preprocesamiento para predecir el RUL con precisión.

```

def prepare_train_data(data, factor = 0):
    # Hacer una copia del DataFrame original para evitar cambios en el
    DataFrame original

```

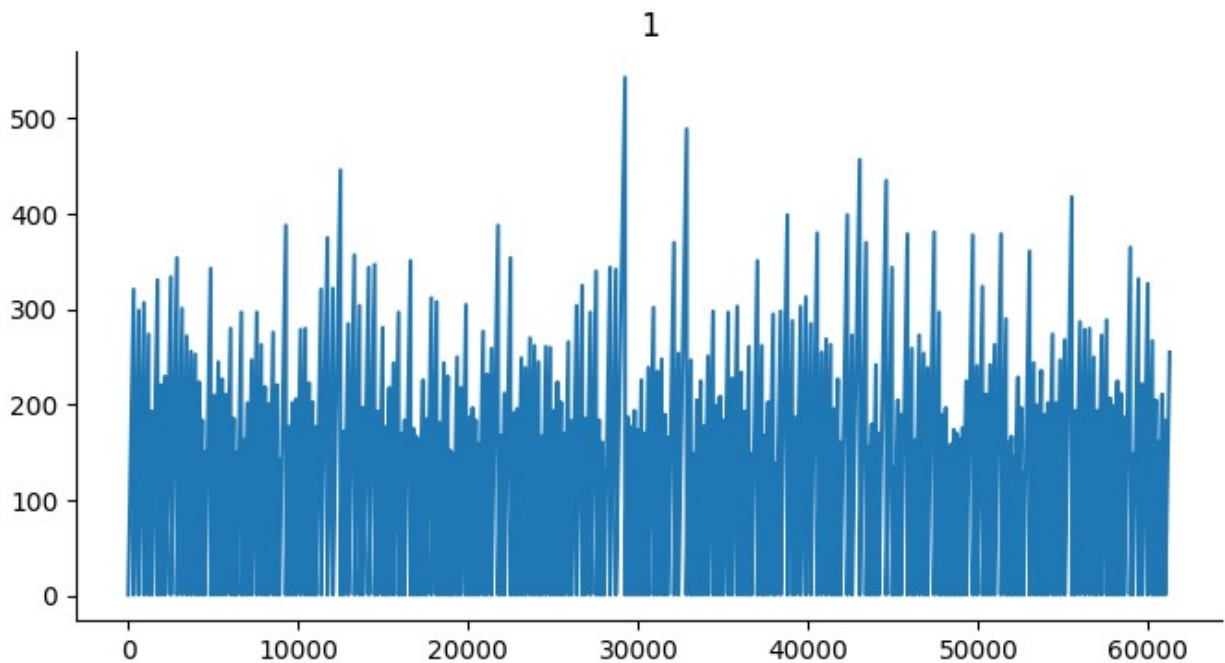
```
df = data.copy()
#Agrupar por
```

##Análisis experimento 4

fd_004

```
{"type": "dataframe", "variable_name": "fd_004"}
```

```
from matplotlib import pyplot as plt
fd_004[1].plot(kind='line', figsize=(8, 4), title=1)
plt.gca().spines[['top', 'right']].set_visible(False)
```



##Análisis del grafico

El gráfico muestra la alta variabilidad de un sensor en el dataset FD004, con más de 60.000 registros. Se observan muchos picos y caídas bruscas, lo que indica un comportamiento complejo y ruidoso típico de este conjunto, útil para detectar fallas pero que requiere filtrado para un análisis efectivo.

##Conclusion FD001 Degradación consistente en entorno controlado. Tendencias claras en sensores (T50, P30). Ideal para análisis en condiciones estables.

FD002 Variabilidad por condiciones operativas. Dispersión en gráficos y sensores. Saltos en Nc y TRA por cambios de condición. Escenario realista.

FD003 Múltiples fallas simultáneas. Densidad al final del gráfico. Modelos de pronóstico complejos.

FD004 Condiciones variables, uso creciente. Influencia de factores externos en sensores. Escenario realista complejo.

Comparación FD001: Controlado. FD002/FD004: Realista/variable. FD003: Fallas múltiples. Gráfico según objetivo.

Conclusión Visualización de la degradación. Información valiosa para mantenimiento predictivo. Perspectivas únicas por conjunto y gráfico.