

December 31, 2024

# 1 Descripción de las columnas del DataFrame

## 1.1 1. Información básica

1. **ROW\_NUMBER:** Número de fila o registro único.
  2. **MISSION\_NUMBER:** Número de misión asignado.
  3. **OP\_ORDER:** Orden operacional asociada a la misión.
  4. **UNIT:** Unidad militar encargada de la misión.
  5. **AIRCRAFT\_TYPE\_MDS:** Tipo o modelo del avión utilizado en la misión.
- 

## 1.2 2. Detalles de aeronaves

6. **NBR\_ATTACK\_EFFEC\_AIRCRAFT:** Número de aeronaves efectivas en el ataque.
  7. **SORTIE\_DUPE:** Duplicado de salidas (para validar datos).
  8. **NBR\_ABORT\_AIRCRAFT:** Número de aeronaves que abortaron la misión.
  9. **NBR\_LOST\_AIRCRAFT:** Número de aeronaves perdidas durante la misión.
- 

## 1.3 3. Información del objetivo

10. **TARGET\_NAME:** Nombre del objetivo.
  11. **TGT\_TYPE:** Tipo de objetivo (por ejemplo, infraestructura militar, ciudad, etc.).
  12. **SOURCE\_UTM\_JAPAN\_B:** Coordenadas UTM del origen del objetivo en Japón.
  13. **SOURCE\_TGT\_UTM:** Coordenadas UTM del origen del objetivo en otro sistema.
  14. **TGT\_MGRS:** Referencia del objetivo en sistema MGRS.
  15. **TGT\_LATITUDE\_WGS84:** Latitud del objetivo (sistema WGS84).
  16. **TGT\_LONGITUDE\_WGS84:** Longitud del objetivo (sistema WGS84).
  17. **SOURCE\_TGT\_LAT:** Latitud de la fuente de lanzamiento.
  18. **SOURCE\_TGT\_LONG:** Longitud de la fuente de lanzamiento.
- 

## 1.4 4. Armamento

19. **NBR\_OF\_WEAPONS:** Número de armas utilizadas en la misión.
20. **WEAPONS\_TYPE:** Tipo de armas o municiones empleadas.
21. **BOMB\_SIGHTING\_METHOD:** Método de avistamiento para bombardeo.
22. **TOTAL\_BOMBLOAD\_IN\_LBS:** Carga total de bombas en libras.

---

## 1.5 5. Detalles de la misión

- 23. **TOT:** Tiempo total de la misión.
  - 24. **MISSION\_TYPE:** Tipo de misión (por ejemplo, ataque, reconocimiento, etc.).
  - 25. **ALTITUDE\_FT:** Altitud de vuelo en pies.
  - 26. **CALLSIGN:** Indicativo de la misión.
  - 27. **BDA:** Evaluación de daños en batalla (Battle Damage Assessment).
  - 28. **NOSE\_FUZE:** Tipo de fusible de la nariz de la bomba.
  - 29. **TAIL\_FUZE:** Tipo de fusible de la cola de la bomba.
  - 30. **CALCULATED\_BOMBLOAD\_LBS:** Carga calculada de bombas en libras.
- 

## 1.6 6. Información adicional

- 31. **RECORD\_SOURCE:** Fuente de registro de la misión.
  - 32. **DAY:** Día de la misión.
  - 33. **MONTH:** Mes de la misión.
  - 34. **YEAR:** Año de la misión.
  - 35. **MISSION\_DATE:** Fecha completa de la misión.
  - 36. **City/Town:** Ciudad o localidad relacionada con la misión.
  - 37. **geometry:** Representación geográfica del objetivo.
  - 38. **In\_Land:** Indicador de si el objetivo está en tierra firme o en el agua.
- 

**Nota:** Si necesitas aclaraciones adicionales o realizar un análisis específico, ¡avísame!

## 2

---

## 3 IMPORTAR LIBRERIAS

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore", category=pd.errors.SettingWithCopyWarning)
```

## 4

---

## 5 LECTURA BASES DE DATOS

```
[2]: file_path = './data/df_bombardeos_procesado.csv'
df_bombardeos_51_52 = pd.read_csv(file_path, sep=',')
df_bombardeos_51_52.head()
```

```
[2]:
```

	ROW_NUMBER	MISSION_NUMBER	OP_ORDER	UNIT	AIRCRAFT_TYPE_MDS	\
0	2	433	174-51	98th Bomb Wing	B-29	
1	3	433	174-51	307th Bomb Wing	B-29	
2	4	433	174-51	307th Bomb Wing	B-29	
3	5	433	174-51	98th Bomb Wing	B-29	
4	6	433	174-51	98th Bomb Wing	B-29	

	NBR_ATTACK_EFFEC_AIRCRAFT	SORTIE_DUPE	NBR_ABORT_AIRCRAFT	\
0	1.0	NaN	NaN	
1	NaN	1.0	NaN	
2	1.0	1.0	NaN	
3	1.0	NaN	NaN	
4	1.0	NaN	NaN	

	NBR_LOST_AIRCRAFT	TARGET_NAME	...	TAIL_FUZE	CALCULATED_BOMBLOAD_LBS	\
0	NaN	Changdo-ri	...	Non-delay	12000.0	
1	NaN	NaN	...	Non-delay	4000.0	
2	NaN	NaN	...	Non-delay	8000.0	
3	NaN	Anju	...	Non-delay	16000.0	
4	NaN	Hamhung	...	Non-delay	16000.0	

	RECORD_SOURCE	DAY	MONTH	YEAR	MISSION_DATE	City/Town	\
0	EXETER	1	6	1951	1951-06-01	Changdo-ri	
1	EXETER	1	6	1951	1951-06-01	North Korea	
2	EXETER	1	6	1951	1951-06-01	Seosam-ri	
3	EXETER	1	6	1951	1951-06-01	Anju-si	
4	EXETER	1	6	1951	1951-06-01	Hamhung-si	

	geometry	In_Land
0	POINT (127.66974 38.49881)	True
1	POINT (126.78196876190476 38.93594352380953)	True
2	POINT (125.50653 39.66879)	True
3	POINT (125.66717 39.60215)	True
4	POINT (127.56091 39.91217)	True

[5 rows x 38 columns]

```
[3]: # Número total de misiones
total_missions = df_bombardeos_51_52['MISSION_NUMBER'].nunique()
print(f"Total de misiones: {total_missions}")
```

```

# Misiones por tipo
missions_by_type = df_bombardeos_51_52['MISSION_TYPE'].value_counts()
print("\nMisiones por tipo:")
print(missions_by_type)

# Duración promedio de las misiones
# Primero, convertimos la columna 'TOT' a un tipo numérico, ignorando errores
df_bombardeos_51_52['TOT'] = pd.to_numeric(df_bombardeos_51_52['TOT'],
    ↪errors='coerce')
average_duration = df_bombardeos_51_52['TOT'].mean()
print(f"\nDuración promedio de las misiones: {average_duration:.2f}")

# Altitud promedio de las misiones por tipo
# Primero, convertimos la columna 'ALTITUDE_FT' a un tipo numérico, ignorando
    ↪errores
df_bombardeos_51_52['ALTITUDE_FT'] = pd.
    ↪to_numeric(df_bombardeos_51_52['ALTITUDE_FT'], errors='coerce')
average_altitude_by_type = df_bombardeos_51_52.
    ↪groupby('MISSION_TYPE')['ALTITUDE_FT'].mean()
print("\nAltitud promedio de las misiones por tipo:")
print(average_altitude_by_type)

```

Total de misiones: 670

Misiones por tipo:

MISSION_TYPE	
Interdiction	3657
Close support	2708
Psychological Warfare	1993
Recon	937
Interdiciton	268
Combat crew evaluation	122
Shoran evaluation	67
Golfball evaluation	42
Experimental drop on railroad tracks	25
MSQ-1 evaluation	20
Close Support	18
Shoran training	10
Maximum Effort Attack on Pyongyang	8
Golfball evalutation	7
Attack on Rashin	6
close support	6
Evaluate combat crew	6
BDA photo	4
Evaluation mission	3
Golfball evaulation	3
ECM mission	2

Interdicton	2
Interdcition	2
Shoran evaulation	2
Interduction	2
Night Interdiction and Close Support	1
Experimental Drops on Railroad Tracks	1
Experimental drop	1
X-port	1
ECM	1

Name: count, dtype: int64

Duración promedio de las misiones: 97660.60

Altitud promedio de las misiones por tipo:

MISSION_TYPE	
Attack on Rashin	NaN
BDA photo	NaN
Close Support	19416.666667
Close support	17840.047265
Combat crew evaluation	20539.709302
ECM	24000.000000
ECM mission	NaN
Evaluate combat crew	20400.000000
Evaluation mission	NaN
Experimental Drops on Railroad Tracks	NaN
Experimental drop	300.000000
Experimental drop on railroad tracks	250.000000
Golfball evaluation	17960.937500
Golfball evalutation	19550.000000
Golfball evaulation	18000.000000
Interdcition	NaN
Interdiciton	20935.057471
Interdiction	20373.953052
Interdicton	NaN
Interduction	NaN
MSQ-1 evaluation	24750.000000
Maximum Effort Attack on Pyongyang	NaN
Night Interdiction and Close Support	NaN
Psychological Warfare	21499.787234
Recon	20161.256831
Shoran evaluation	20076.190476
Shoran evaulation	19800.000000
Shoran training	21100.000000
X-port	NaN
close support	17650.000000

Name: ALTITUDE\_FT, dtype: float64

```
[4]: # Número total de aeronaves enviadas
total_effective_aircraft = df_bombardeos_51_52['NBR_ATTACK_EFFEC_AIRCRAFT'].
    ↪sum()
print(f"Número total de aeronaves efectivas: {total_effective_aircraft}")

# Tasa de abortos de misión
# Primero, eliminamos los valores NaN para evitar errores en la división
df_bombardeos_51_52['NBR_ABORT_AIRCRAFT'] =
    ↪df_bombardeos_51_52['NBR_ABORT_AIRCRAFT'].fillna(0)
df_bombardeos_51_52['NBR_ATTACK_EFFEC_AIRCRAFT'] =
    ↪df_bombardeos_51_52['NBR_ATTACK_EFFEC_AIRCRAFT'].fillna(0)
mission_abort_rate = df_bombardeos_51_52['NBR_ABORT_AIRCRAFT'].sum() /
    ↪total_effective_aircraft
print(f"Tasa de abortos de misión: {mission_abort_rate:.2%}")

# Pérdidas de aeronaves por misión
# Primero, convertimos la columna 'NBR_LOST_AIRCRAFT' a un tipo numérico,
    ↪ignorando errores
df_bombardeos_51_52['NBR_LOST_AIRCRAFT'] = pd.
    ↪to_numeric(df_bombardeos_51_52['NBR_LOST_AIRCRAFT'], errors='coerce')
average_aircraft_losses = df_bombardeos_51_52['NBR_LOST_AIRCRAFT'].mean()
print(f"Pérdidas promedio de aeronaves por misión: {average_aircraft_losses:.
    ↪2f}")

# Tipos de misión con mayores pérdidas
losses_by_mission_type = df_bombardeos_51_52.
    ↪groupby('MISSION_TYPE')['NBR_LOST_AIRCRAFT'].mean().
    ↪sort_values(ascending=False)
print("\nTipos de misión con mayores pérdidas de aeronaves:")
print(losses_by_mission_type)

# Eficiencia de las misiones
df_bombardeos_51_52['efficiency'] =
    ↪df_bombardeos_51_52['NBR_ATTACK_EFFEC_AIRCRAFT'] /
    ↪(df_bombardeos_51_52['NBR_ATTACK_EFFEC_AIRCRAFT'] +
    ↪df_bombardeos_51_52['NBR_ABORT_AIRCRAFT'])
average_efficiency = df_bombardeos_51_52['efficiency'].mean()
print(f"\nEficiencia promedio de las misiones: {average_efficiency:.2%}")
```

Número total de aeronaves efectivas: 28198.0

Tasa de abortos de misión: 1.58%

Pérdidas promedio de aeronaves por misión: 1.10

Tipos de misión con mayores pérdidas de aeronaves:

MISSION_TYPE	
Interdiction	1.285714
Psychological Warfare	1.000000

Recon	1.000000
Attack on Rashin	NaN
BDA photo	NaN
Close Support	NaN
Close support	NaN
Combat crew evaluation	NaN
ECM	NaN
ECM mission	NaN
Evaluate combat crew	NaN
Evaluation mission	NaN
Experimental Drops on Railroad Tracks	NaN
Experimental drop	NaN
Experimental drop on railroad tracks	NaN
Golfball evaluation	NaN
Golfball evalutation	NaN
Golfball evaulation	NaN
Interdcition	NaN
Interdiciton	NaN
Interdicton	NaN
Interduction	NaN
MSQ-1 evaluation	NaN
Maximum Effort Attack on Pyongyang	NaN
Night Interdiction and Close Support	NaN
Shoran evaluation	NaN
Shoran evaulation	NaN
Shoran training	NaN
X-port	NaN
close support	NaN

Name: NBR\_LOST\_AIRCRAFT, dtype: float64

Eficiencia promedio de las misiones: 98.56%

```
[5]: # Frecuencia de tipos de objetivos
target_type_frequency = df_bombardeos_51_52['TGT_TYPE'].value_counts()
print("Frecuencia de tipos de objetivos:")
print(target_type_frequency)

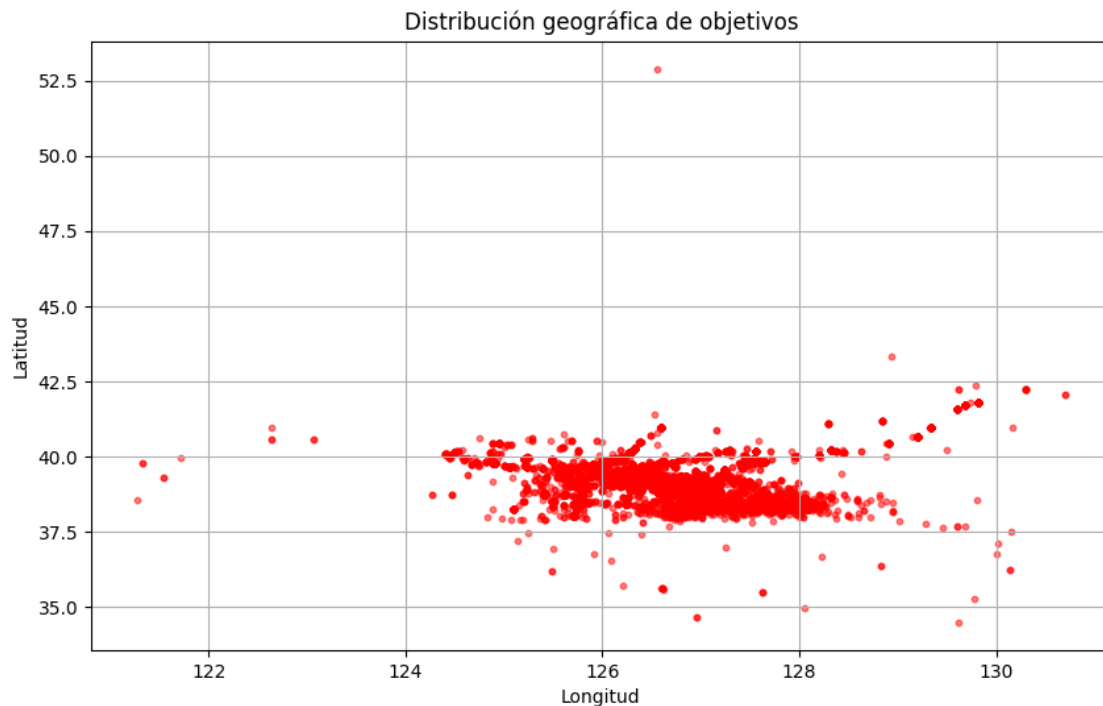
# Distribución geográfica de objetivos
plt.figure(figsize=(10, 6))
plt.scatter(df_bombardeos_51_52['TGT_LONGITUDE_WGS84'],
            df_bombardeos_51_52['TGT_LATITUDE_WGS84'], alpha=0.5, c='red', s=10)
plt.title('Distribución geográfica de objetivos')
plt.xlabel('Longitud')
plt.ylabel('Latitud')
plt.grid(True)
plt.show()
```

```
# Eficiencia por tipo de objetivo
efficiency_by_target_type = df_bombardeos_51_52.
    ↳groupby('TGT_TYPE')['efficiency'].mean()
print("\nEficiencia por tipo de objetivo:")
print(efficiency_by_target_type)
```

Frecuencia de tipos de objetivos:

TGT_TYPE	
Marshalling yard	1302
Unknown target	1005
Jettisoned	943
Troop concentration	638
Airfield	331
...	
"Personnel shelters, supply route, and assembly area"	1
"Personnel shelters, supply point, and assembly area"	1
Yonghung	1
"Personnel and supply shelters, supply caves, and bunkers"	1
Command post headquarters	1

Name: count, Length: 421, dtype: int64



Eficiencia por tipo de objetivo:

TGT\_TYPE



```

""A""
1.0
"8th North Korean Division Headquarters, supply and personnel shelters, and a
bridge" 1.0
"Area ""A""
1.0
"Area ""E""
1.0
"Artillery positions, large bunkers, and supply areas"
1.0
...
Yonghung
1.0
railroad bridge
1.0
returned
1.0
supply and personnel shelters
1.0
supply and personnel shelters and large bunkers
1.0
Name: efficiency, Length: 421, dtype: float64

```

```

[6]: # Promedio de armas utilizadas por misión
df_bombardeos_51_52['NBR_OF_WEAPONS'] = pd.
    ↳to_numeric(df_bombardeos_51_52['NBR_OF_WEAPONS'], errors='coerce')
average_weapons_per_mission = df_bombardeos_51_52['NBR_OF_WEAPONS'].mean()
print(f"Promedio de armas utilizadas por misión: {average_weapons_per_mission:.
    ↳2f}")

# Carga promedio de bombas por misión
average_bombload_per_mission = df_bombardeos_51_52['TOTAL_BOMBLOAD_IN_LBS'].
    ↳mean()
print(f"Carga promedio de bombas por misión: {average_bombload_per_mission:.
    ↳2f}")

# Uso de tipos de armas
weapons_type_frequency = df_bombardeos_51_52['WEAPONS_TYPE'].value_counts()
print("Frecuencia de tipos de armas:")
print(weapons_type_frequency)

```

```

Promedio de armas utilizadas por misión: 47.29
Carga promedio de bombas por misión: 37600.00
Frecuencia de tipos de armas:
WEAPONS_TYPE
500 GP          5209
M16-A1 Leaflet Bomb  2015

```

M-46	1046
100 GP	991
M-120	254
1000 GP	252
2000 GP	29
M-17 500 GP	25
2000 SAP	15
M-89	12
1000 Razon	11
500 M-76	11
M-20 (E-48)	10
500 DP	9
Unknown	9
250 GP	9
Incendiary	7
M146	6
500 M-20E48 I	6
M-60	4
500 M20 E48	3
500 M-20E48	3
M-152	3
M-90	3
500 GP M-23 A bombs	2
M-76	2
Incendiary cluster	2
500 M-20 E48	2
E-48 M-20	2
100 lb M46 M-46	1
500 G0	1
200 SAP	1
M-16A1	1
E-48 M-90	1
M-12	1

Name: count, dtype: int64

```
[8]: # Agrupar por mes y calcular métricas clave
monthly_metrics = df_bombardeos_51_52.groupby('MONTH').agg({
    'efficiency': 'mean',
    'NBR_LOST_AIRCRAFT': 'sum',
    'NBR_ABORT_AIRCRAFT': 'sum',
    'MISSION_NUMBER': 'count'
}).reset_index()

# Graficar misiones y eficiencia promedio por mes
fig, ax1 = plt.subplots(figsize=(14, 7))

ax1.set_xlabel('Mes')
```

```

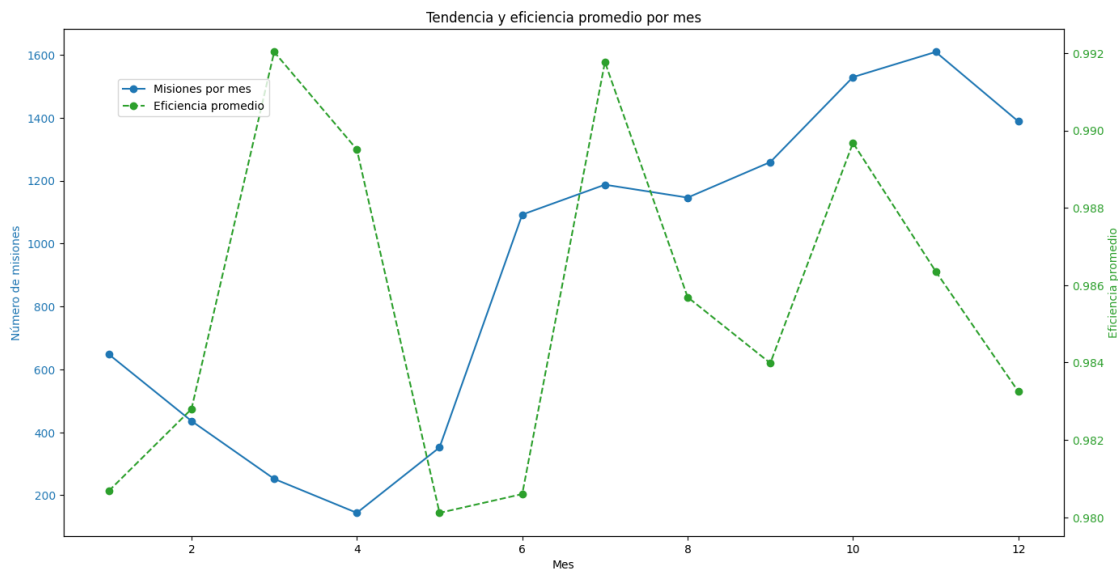
ax1.set_ylabel('Número de misiones', color='tab:blue')
ax1.plot(monthly_metrics['MONTH'], monthly_metrics['MISSION_NUMBER'],
        ↪marker='o', linestyle='-', color='tab:blue', label='Misiones por mes')
ax1.tick_params(axis='y', labelcolor='tab:blue')

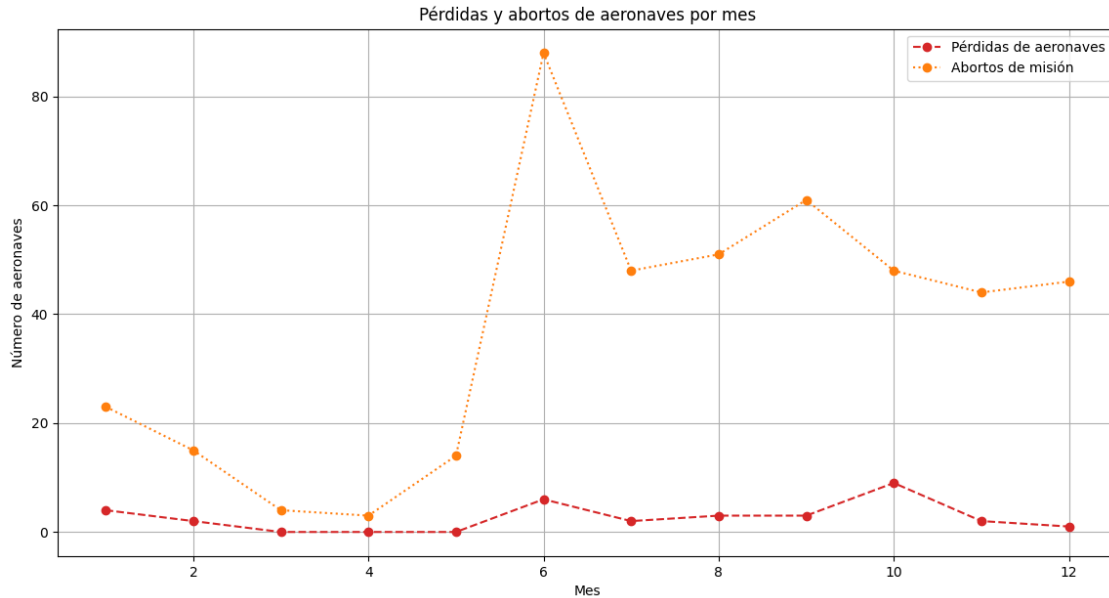
ax2 = ax1.twinx()
ax2.set_ylabel('Eficiencia promedio', color='tab:green')
ax2.plot(monthly_metrics['MONTH'], monthly_metrics['efficiency'], marker='o',
        ↪linestyle='--', color='tab:green', label='Eficiencia promedio')
ax2.tick_params(axis='y', labelcolor='tab:green')

fig.tight_layout()
fig.legend(loc='upper left', bbox_to_anchor=(0.1, 0.9))
plt.title('Tendencia y eficiencia promedio por mes')
plt.show()

# Pérdidas y abortos por mes
plt.figure(figsize=(14, 7))
plt.plot(monthly_metrics['MONTH'], monthly_metrics['NBR_LOST_AIRCRAFT'],
        ↪marker='o', linestyle='--', color='tab:red', label='Pérdidas de aeronaves')
plt.plot(monthly_metrics['MONTH'], monthly_metrics['NBR_ABORT_AIRCRAFT'],
        ↪marker='o', linestyle=':', color='tab:orange', label='Abortos de misión')
plt.title('Pérdidas y abortos de aeronaves por mes')
plt.xlabel('Mes')
plt.ylabel('Número de aeronaves')
plt.legend()
plt.grid(True)
plt.show()

```





```
[10]: # Tasa de daños en objetivos
bda_success_rate = df_bombardeos_51_52['BDA'].value_counts(normalize=True)
print("Tasa de daños en objetivos (BDA):")
print(bda_success_rate)

# Relación entre carga de bombas y éxito
df_bombardeos_51_52['TOTAL_BOMBLOAD_IN_LBS'] = pd.
    ↳to_numeric(df_bombardeos_51_52['TOTAL_BOMBLOAD_IN_LBS'], errors='coerce')
df_bombardeos_51_52['BDA_numeric'] = pd.to_numeric(df_bombardeos_51_52['BDA'],
    ↳errors='coerce')
bombload_success_correlation = df_bombardeos_51_52[['TOTAL_BOMBLOAD_IN_LBS',
    ↳'BDA_numeric']].dropna().corr()
print("\nCorrelación entre carga de bombas y éxito (BDA):")
print(bombload_success_correlation)

# Relación entre altura de vuelo y efectividad
altitude_success_correlation = df_bombardeos_51_52[['ALTITUDE_FT',
    ↳'BDA_numeric']].dropna().corr()
print("\nCorrelación entre altura de vuelo y éxito (BDA):")
print(altitude_success_correlation)
```

Tasa de daños en objetivos (BDA):

BDA

Fuzed to discharge leaflets 1000 ft above the terrain.

0.114970

Good results.

0.067216

Mission executed precisely as planned. Fuzed to discharge leaflets 1000 ft above the terrain. Results unobserved due to darkness. 0.055389  
 Fuzed to discharge leaflets 1000 ft above the terrain. Results unobserved due to darkness. 0.043263  
 Excellent results.  
 0.025000

...

3 unknown frontline targets attacked. 2 bombs were removed from the bomb bay prior to take-off. 0.000150  
 Primary target not attacked as the front bomb bay doors would not open over the area. 0.000150  
 1 aircraft due to a bomb rack malfunction dropped half it's bomb load on an unidentified village at YD 1594 0.000150  
 Ground controller reported direct hits on the target.  
 0.000150  
 7/10 to 8/10 cloud coverage prevailed. Unobserved results due to weather conditions. 0.000150  
 Name: proportion, Length: 2044, dtype: float64

Correlación entre carga de bombas y éxito (BDA):

	TOTAL_BOMBLOAD_IN_LBS	BDA_numeric
TOTAL_BOMBLOAD_IN_LBS	NaN	NaN
BDA_numeric	NaN	NaN

Correlación entre altura de vuelo y éxito (BDA):

	ALTITUDE_FT	BDA_numeric
ALTITUDE_FT	NaN	NaN
BDA_numeric	NaN	NaN

```
[12]: # Agrupar por 'City/Town' y contar los objetivos más comunes
common_targets_by_city = df_bombardeos_51_52['City/Town'].value_counts()
print("Ubicación de objetivos más frecuentes:")
print(common_targets_by_city.head(10))

# Visualizar las ubicaciones de lanzamiento y los objetivos en mapas de calor
import folium
from folium.plugins import HeatMap

# Convertir coordenadas a valores numéricos
def convert_to_float(coord):
    try:
        return float(coord)
    except ValueError:
        return None

df_bombardeos_51_52['SOURCE_TGT_LAT'] = df_bombardeos_51_52['SOURCE_TGT_LAT'].
    .apply(convert_to_float)
```

```

df_bombardeos_51_52['SOURCE_TGT_LONG'] = df_bombardeos_51_52['SOURCE_TGT_LONG'].
    ↪apply(convert_to_float)

# Crear un mapa base
mapa = folium.Map(location=[38.5, 127], zoom_start=6)

# Agregar ubicaciones de lanzamiento como un mapa de calor
heat_data_launch = [[row['SOURCE_TGT_LAT'], row['SOURCE_TGT_LONG']] for index,
    ↪row in df_bombardeos_51_52.dropna(subset=['SOURCE_TGT_LAT',
    ↪'SOURCE_TGT_LONG']).iterrows()]
HeatMap(heat_data_launch, radius=10, max_zoom=13).add_to(mapa)

# Agregar ubicaciones de objetivos como un mapa de calor
heat_data_target = [[row['TGT_LATITUDE_WGS84'], row['TGT_LONGITUDE_WGS84']] for
    ↪index, row in df_bombardeos_51_52.dropna(subset=['TGT_LATITUDE_WGS84',
    ↪'TGT_LONGITUDE_WGS84']).iterrows()]
HeatMap(heat_data_target, radius=10, max_zoom=13).add_to(mapa)

# Mostrar el mapa
mapa

# Analizar misiones terrestres vs marítimas
land_missions = df_bombardeos_51_52[df_bombardeos_51_52['In_Land'] == True]
water_missions = df_bombardeos_51_52[df_bombardeos_51_52['In_Land'] == False]

# Comparar métricas clave
land_missions_metrics = land_missions.agg({
    'efficiency': 'mean',
    'NBR_LOST_AIRCRAFT': 'sum',
    'NBR_ABORT_AIRCRAFT': 'sum',
    'MISSION_NUMBER': 'count'
})

water_missions_metrics = water_missions.agg({
    'efficiency': 'mean',
    'NBR_LOST_AIRCRAFT': 'sum',
    'NBR_ABORT_AIRCRAFT': 'sum',
    'MISSION_NUMBER': 'count'
})

print("\nMétricas de misiones terrestres:")
print(land_missions_metrics)

print("\nMétricas de misiones marítimas:")
print(water_missions_metrics)

```

Ubicación de objetivos más frecuentes:

```

City/Town
North Korea      1756
Yangdok County   706
Hamhung-si       657
P'yŏngyang       370
Songnim-si       319
Nampo            299
Kowon County     280
Sunchon-si       249
Kaesong          217
Anju-si          201
Name: count, dtype: int64

```

```

Métricas de misiones terrestres:
efficiency        0.985598
NBR_LOST_AIRCRAFT 31.000000
NBR_ABORT_AIRCRAFT 439.000000
MISSION_NUMBER    10859.000000
dtype: float64

```

```

Métricas de misiones marítimas:
efficiency        0.985732
NBR_LOST_AIRCRAFT 1.000000
NBR_ABORT_AIRCRAFT 6.000000
MISSION_NUMBER    183.000000
dtype: float64

```

```

[13]: # Verificar consistencia entre CALCULATED_BOMBLOAD_LBS y TOTAL_BOMBLOAD_IN_LBS
df_bombardeos_51_52['CALCULATED_BOMBLOAD_LBS'] = pd.
↳to_numeric(df_bombardeos_51_52['CALCULATED_BOMBLOAD_LBS'], errors='coerce')
df_bombardeos_51_52['TOTAL_BOMBLOAD_IN_LBS'] = pd.
↳to_numeric(df_bombardeos_51_52['TOTAL_BOMBLOAD_IN_LBS'], errors='coerce')
bombload_consistency = df_bombardeos_51_52[['CALCULATED_BOMBLOAD_LBS',
↳'TOTAL_BOMBLOAD_IN_LBS']].dropna()
consistency_check = bombload_consistency['CALCULATED_BOMBLOAD_LBS'] ==
↳bombload_consistency['TOTAL_BOMBLOAD_IN_LBS']
consistency_rate = consistency_check.mean()
print(f"Consistencia entre CALCULATED_BOMBLOAD_LBS y TOTAL_BOMBLOAD_IN_LBS:
↳{consistency_rate:.2%}")

# Comparar métodos de avistamiento con el éxito medido por BDA
bda_success_by_sighting_method = df_bombardeos_51_52.
↳groupby('BOMB_SIGHTING_METHOD')['BDA_numeric'].mean()
print("\nÉxito medido por BDA según el método de avistamiento:")
print(bda_success_by_sighting_method)

# Correlaciones clave

```

```

# NBR_OF_WEAPONS vs. BDA
weapons_bda_correlation = df_bombardeos_51_52[['NBR_OF_WEAPONS',
↪ 'BDA_numeric']].dropna().corr()
print("\nCorrelación entre NBR_OF_WEAPONS y BDA:")
print(weapons_bda_correlation)

# ALTITUDE_FT vs. NBR_LOST_AIRCRAFT
altitude_losses_correlation = df_bombardeos_51_52[['ALTITUDE_FT',
↪ 'NBR_LOST_AIRCRAFT']].dropna().corr()
print("\nCorrelación entre ALTITUDE_FT y NBR_LOST_AIRCRAFT:")
print(altitude_losses_correlation)

```

Consistencia entre CALCULATED\_BOMBLOAD\_LBS y TOTAL\_BOMBLOAD\_IN\_LBS: 100.00%

Éxito medido por BDA según el método de avistamiento:

BOMB\_SIGHTING\_METHOD

MPQ-2	NaN
Radar	NaN
Radar and visual	NaN
Shoran radar	NaN
Shoran radar and Radar	NaN
Shoran radar and Visual	NaN
Shoran radar and radar	NaN
Shoran radar and visual	NaN
Shoran radar and voice signal	NaN
Visual	NaN
Visual and Shoran radar	NaN
Visual and radar	NaN
Visual with radar assistance	NaN

Name: BDA\_numeric, dtype: float64

Correlación entre NBR\_OF\_WEAPONS y BDA:

	NBR_OF_WEAPONS	BDA_numeric
NBR_OF_WEAPONS	NaN	NaN
BDA_numeric	NaN	NaN

Correlación entre ALTITUDE\_FT y NBR\_LOST\_AIRCRAFT:

	ALTITUDE_FT	NBR_LOST_AIRCRAFT
ALTITUDE_FT	1.0	NaN
NBR_LOST_AIRCRAFT	NaN	NaN