

Tarea1_Santana_Abasolo

April 30, 2025

```
[ ]: #Importamos librerias
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

1. Cargar la base de datos en el ambiente. Identifique los tipos de datos que se encuentran en la base, realice estadísticas descriptivas sobre las variables importantes (Hint: Revisar la distribuciones, datos faltantes, outliers, etc.) y limpie las variables cuando sea necesario

```
[ ]: #Leemos y visualizamos la base de datos
df = pd.read_csv("../data/machine_failure_data.csv")
df
```

```
[ ]:
```

	Date	Location	Min_Temp	Max_Temp	Leakage	Evaporation	\
0	12/1/2008	3	13.4	22.9	0.6	NaN	
1	12/2/2008	3	7.4	25.1	0.0	NaN	
2	12/3/2008	3	12.9	25.7	0.0	NaN	
3	12/4/2008	3	9.2	28.0	0.0	NaN	
4	12/5/2008	3	17.5	32.3	1.0	NaN	
...	
142188	6/20/2017	42	3.5	21.8	0.0	NaN	
142189	6/21/2017	42	2.8	23.4	0.0	NaN	
142190	6/22/2017	42	3.6	25.3	0.0	NaN	
142191	6/23/2017	42	5.4	26.9	0.0	NaN	
142192	6/24/2017	42	7.8	27.0	0.0	NaN	

	Electricity	Parameter1_Dir	Parameter1_Speed	Parameter2_9am	...	\
0	NaN	W	44.0	W	...	
1	NaN	WNW	44.0	NNW	...	
2	NaN	WSW	46.0	W	...	
3	NaN	NE	24.0	SE	...	
4	NaN	W	41.0	ENE	...	
...	
142188	NaN	E	31.0	ESE	...	
142189	NaN	E	31.0	SE	...	

142190	NaN	NNW	22.0	SE ...
142191	NaN	N	37.0	SE ...
142192	NaN	SE	28.0	SSE ...

	Parameter3_3pm	Parameter4_9am	Parameter4_3pm	Parameter5_9am	\
0	24.0	71.0	22.0	1007.7	
1	22.0	44.0	25.0	1010.6	
2	26.0	38.0	30.0	1007.6	
3	9.0	45.0	16.0	1017.6	
4	20.0	82.0	33.0	1010.8	
...	
142188	13.0	59.0	27.0	1024.7	
142189	11.0	51.0	24.0	1024.6	
142190	9.0	56.0	21.0	1023.5	
142191	9.0	53.0	24.0	1021.0	
142192	7.0	51.0	24.0	1019.4	

	Parameter5_3pm	Parameter6_9am	Parameter6_3pm	Parameter7_9am	\
0	1007.1	8.0	NaN	16.9	
1	1007.8	NaN	NaN	17.2	
2	1008.7	NaN	2.0	21.0	
3	1012.8	NaN	NaN	18.1	
4	1006.0	7.0	8.0	17.8	
...	
142188	1021.2	NaN	NaN	9.4	
142189	1020.3	NaN	NaN	10.1	
142190	1019.1	NaN	NaN	10.9	
142191	1016.8	NaN	NaN	12.5	
142192	1016.5	3.0	2.0	15.1	

	Parameter7_3pm	Failure_today
0	21.8	No
1	24.3	No
2	23.2	No
3	26.5	No
4	29.7	No
...
142188	20.9	No
142189	22.4	No
142190	24.5	No
142191	26.1	No
142192	26.0	No

[142193 rows x 22 columns]

```
[3]: #Visualizamos la información de los datos del df
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142193 entries, 0 to 142192
Data columns (total 22 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date                  142193 non-null object
1   Location              142193 non-null int64
2   Min_Temp              141556 non-null float64
3   Max_Temp              141871 non-null float64
4   Leakage               140787 non-null float64
5   Evaporation           81350 non-null float64
6   Electricity           74377 non-null float64
7   Parameter1_Dir        132863 non-null object
8   Parameter1_Speed      132923 non-null float64
9   Parameter2_9am        132180 non-null object
10  Parameter2_3pm        138415 non-null object
11  Parameter3_9am        140845 non-null float64
12  Parameter3_3pm        139563 non-null float64
13  Parameter4_9am        140419 non-null float64
14  Parameter4_3pm        138583 non-null float64
15  Parameter5_9am        128179 non-null float64
16  Parameter5_3pm        128212 non-null float64
17  Parameter6_9am        88536 non-null float64
18  Parameter6_3pm        85099 non-null float64
19  Parameter7_9am        141289 non-null float64
20  Parameter7_3pm        139467 non-null float64
21  Failure_today         140787 non-null object
dtypes: float64(16), int64(1), object(5)
memory usage: 23.9+ MB

```

Parameter6_9am tiene 88536 datos y Parameter6_3pm 85099, aproximadamente 51 000 datos menos en comparación a las demás variables por lo tanto los eliminamos directamente. Eliminamos así mismo a Evaporation y Electricity. En el caso de Leakage resulta ser un estimador perfecto para el modelo, por lo que también lo eliminamos.

```

[4]: #Eliminamos las columnas y volvemos a visualizar los datos del df
df=df.drop(columns=["Parameter6_9am","Parameter6_3pm"])
df=df.drop(columns=["Evaporation","Electricity"])
df=df.drop(columns=["Leakage"])
df.info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142193 entries, 0 to 142192
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date                  142193 non-null object
1   Location              142193 non-null int64
2   Min_Temp              141556 non-null float64

```

```

3   Max_Temp          141871 non-null float64
4   Parameter1_Dir    132863 non-null object
5   Parameter1_Speed  132923 non-null float64
6   Parameter2_9am    132180 non-null object
7   Parameter2_3pm    138415 non-null object
8   Parameter3_9am    140845 non-null float64
9   Parameter3_3pm    139563 non-null float64
10  Parameter4_9am    140419 non-null float64
11  Parameter4_3pm    138583 non-null float64
12  Parameter5_9am    128179 non-null float64
13  Parameter5_3pm    128212 non-null float64
14  Parameter7_9am    141289 non-null float64
15  Parameter7_3pm    139467 non-null float64
16  Failure_today     140787 non-null object
dtypes: float64(11), int64(1), object(5)
memory usage: 18.4+ MB

```

```
[5]: df.describe(include='all')
```

```

[5]:
count      Date      Location      Min_Temp      Max_Temp  Parameter1_Dir  \
count      142193  142193.000000  141556.000000  141871.000000      132863
unique      3436      NaN          NaN          NaN          NaN          16
top         6/23/2017      NaN          NaN          NaN          NaN          W
freq         49      NaN          NaN          NaN          NaN          9780
mean        NaN      24.740655      12.186400      23.226784      NaN
std         NaN      14.237503      6.403283      7.117618      NaN
min         NaN      1.000000     -8.500000     -4.800000      NaN
25%         NaN      12.000000      7.600000     17.900000      NaN
50%         NaN      25.000000     12.000000     22.600000      NaN
75%         NaN      37.000000     16.800000     28.200000      NaN
max         NaN      49.000000     33.900000     48.100000      NaN

      Parameter1_Speed  Parameter2_9am  Parameter2_3pm  Parameter3_9am  \
count      132923.000000      132180      138415      140845.000000
unique      NaN          16          16          NaN
top         NaN          N          SE          NaN
freq         NaN      11393      10663          NaN
mean        39.984292      NaN          NaN          14.001988
std         13.588801      NaN          NaN          8.893337
min         6.000000      NaN          NaN          0.000000
25%         31.000000      NaN          NaN          7.000000
50%         39.000000      NaN          NaN          13.000000
75%         48.000000      NaN          NaN          19.000000
max         135.000000      NaN          NaN          130.000000

      Parameter3_3pm  Parameter4_9am  Parameter4_3pm  Parameter5_9am  \
count      139563.000000      140419.000000      138583.000000      128179.000000

```

unique	NaN	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN
mean	18.637576	68.843810	51.482606	1017.653758
std	8.803345	19.051293	20.797772	7.105476
min	0.000000	0.000000	0.000000	980.500000
25%	13.000000	57.000000	37.000000	1012.900000
50%	19.000000	70.000000	52.000000	1017.600000
75%	24.000000	83.000000	66.000000	1022.400000
max	87.000000	100.000000	100.000000	1041.000000

	Parameter5_3pm	Parameter7_9am	Parameter7_3pm	Failure_today
count	128212.000000	141289.000000	139467.000000	140787
unique	NaN	NaN	NaN	2
top	NaN	NaN	NaN	No
freq	NaN	NaN	NaN	109332
mean	1015.258204	16.987509	21.687235	NaN
std	7.036677	6.492838	6.937594	NaN
min	977.100000	-7.200000	-5.400000	NaN
25%	1010.400000	12.300000	16.600000	NaN
50%	1015.200000	16.700000	21.100000	NaN
75%	1020.000000	21.600000	26.400000	NaN
max	1039.600000	40.200000	46.700000	NaN

[6]: *#Aqui pasamos de las 16 direcciones de viento a angulos y posteriormente a 4 grupos (N, E, S y O)*

```

direccion_a_angulo = {
    'N': 0,
    'NNE': 22.5,
    'NE': 45,
    'ENE': 67.5,
    'E': 90,
    'ESE': 112.5,
    'SE': 135,
    'SSE': 157.5,
    'S': 180,
    'SSW': 202.5,
    'SW': 225,
    'WSW': 247.5,
    'W': 270,
    'WNW': 292.5,
    'NW': 315,
    'NNW': 337.5
}

# Mapear a ángulos
df['Parameter1_Dir_angle'] = df['Parameter1_Dir'].map(direccion_a_angulo)

```

```

df['Parameter2_9am_angle'] = df['Parameter2_9am'].map(direccion_a_angulo)
df['Parameter2_3pm_angle'] = df['Parameter2_3pm'].map(direccion_a_angulo)

df['Parameter1_Dir_angle'] = df['Parameter1_Dir_angle'].fillna(0)
df['Parameter2_9am_angle'] = df['Parameter2_9am_angle'].fillna(0)
df['Parameter2_3pm_angle'] = df['Parameter2_3pm_angle'].fillna(0)

def agrupar_direccion(angle):
    if (angle >= 315 or angle < 45):
        return 'N'
    elif (angle >= 45 and angle < 135):
        return 'E'
    elif (angle >= 135 and angle < 225):
        return 'S'
    elif (angle >= 225 and angle < 315):
        return 'W'
    else:
        return 'Desconocido'

columnas_angulos = ['Parameter1_Dir_angle', 'Parameter2_9am_angle',
                    ↪ 'Parameter2_3pm_angle']

# Aplicar la funcion a cada columna que termina en _angle y creamos _region
for col in columnas_angulos:
    nueva_col = col.replace('_angle', '_region')
    df[nueva_col] = df[col].apply(agrupar_direccion)

df=df.
    ↪ drop(columns=["Parameter1_Dir", "Parameter2_9am", "Parameter2_3pm", 'Parameter1_Dir_angle',
    ↪ 'Parameter2_9am_angle', 'Parameter2_3pm_angle'])
df

```

```

[6]:
      Date  Location  Min_Temp  Max_Temp  Parameter1_Speed  \
0    12/1/2008        3     13.4     22.9             44.0
1    12/2/2008        3      7.4     25.1             44.0
2    12/3/2008        3     12.9     25.7             46.0
3    12/4/2008        3      9.2     28.0             24.0
4    12/5/2008        3     17.5     32.3             41.0
...      ...      ...      ...      ...      ...
142188  6/20/2017     42      3.5     21.8             31.0
142189  6/21/2017     42      2.8     23.4             31.0
142190  6/22/2017     42      3.6     25.3             22.0
142191  6/23/2017     42      5.4     26.9             37.0
142192  6/24/2017     42      7.8     27.0             28.0

      Parameter3_9am  Parameter3_3pm  Parameter4_9am  Parameter4_3pm  \
0                20.0             24.0             71.0             22.0

```

1	4.0	22.0	44.0	25.0
2	19.0	26.0	38.0	30.0
3	11.0	9.0	45.0	16.0
4	7.0	20.0	82.0	33.0
...
142188	15.0	13.0	59.0	27.0
142189	13.0	11.0	51.0	24.0
142190	13.0	9.0	56.0	21.0
142191	9.0	9.0	53.0	24.0
142192	13.0	7.0	51.0	24.0

	Parameter5_9am	Parameter5_3pm	Parameter7_9am	Parameter7_3pm	\
0	1007.7	1007.1	16.9	21.8	
1	1010.6	1007.8	17.2	24.3	
2	1007.6	1008.7	21.0	23.2	
3	1017.6	1012.8	18.1	26.5	
4	1010.8	1006.0	17.8	29.7	
...	
142188	1024.7	1021.2	9.4	20.9	
142189	1024.6	1020.3	10.1	22.4	
142190	1023.5	1019.1	10.9	24.5	
142191	1021.0	1016.8	12.5	26.1	
142192	1019.4	1016.5	15.1	26.0	

	Failure_today	Parameter1_Dir_region	Parameter2_9am_region	\
0	No		W	W
1	No		W	N
2	No		W	W
3	No		E	S
4	No		W	E
...
142188	No		E	E
142189	No		E	S
142190	No		N	S
142191	No		N	S
142192	No		S	S

	Parameter2_3pm_region
0	W
1	W
2	W
3	E
4	N
...	...
142188	E
142189	E
142190	N

```
142191          W
142192          N
```

```
[142193 rows x 17 columns]
```

```
[7]: #Transformamos la fecha a formato "datetime" y agrupamos las fechas en 4
      ↪ estaciones
df['Date'] = pd.to_datetime(df['Date'], format='%m/%d/%Y')

def obtener_estacion(fecha):
    mes = fecha.month
    if mes in [12, 1, 2]:
        return 'invierno'
    elif mes in [3, 4, 5]:
        return 'primavera'
    elif mes in [6, 7, 8]:
        return 'verano'
    else:
        return 'otoño'

df['estacion'] = df['Date'].apply(obtener_estacion)
df
```

```
[7]:
```

	Date	Location	Min_Temp	Max_Temp	Parameter1_Speed	\
0	2008-12-01	3	13.4	22.9	44.0	
1	2008-12-02	3	7.4	25.1	44.0	
2	2008-12-03	3	12.9	25.7	46.0	
3	2008-12-04	3	9.2	28.0	24.0	
4	2008-12-05	3	17.5	32.3	41.0	
...	
142188	2017-06-20	42	3.5	21.8	31.0	
142189	2017-06-21	42	2.8	23.4	31.0	
142190	2017-06-22	42	3.6	25.3	22.0	
142191	2017-06-23	42	5.4	26.9	37.0	
142192	2017-06-24	42	7.8	27.0	28.0	

	Parameter3_9am	Parameter3_3pm	Parameter4_9am	Parameter4_3pm	\
0	20.0	24.0	71.0	22.0	
1	4.0	22.0	44.0	25.0	
2	19.0	26.0	38.0	30.0	
3	11.0	9.0	45.0	16.0	
4	7.0	20.0	82.0	33.0	
...	
142188	15.0	13.0	59.0	27.0	
142189	13.0	11.0	51.0	24.0	
142190	13.0	9.0	56.0	21.0	

142191	9.0	9.0	53.0	24.0
142192	13.0	7.0	51.0	24.0

	Parameter5_9am	Parameter5_3pm	Parameter7_9am	Parameter7_3pm	\
0	1007.7	1007.1	16.9	21.8	
1	1010.6	1007.8	17.2	24.3	
2	1007.6	1008.7	21.0	23.2	
3	1017.6	1012.8	18.1	26.5	
4	1010.8	1006.0	17.8	29.7	
...	
142188	1024.7	1021.2	9.4	20.9	
142189	1024.6	1020.3	10.1	22.4	
142190	1023.5	1019.1	10.9	24.5	
142191	1021.0	1016.8	12.5	26.1	
142192	1019.4	1016.5	15.1	26.0	

	Failure_today	Parameter1_Dir_region	Parameter2_9am_region	\
0	No		W	W
1	No		W	N
2	No		W	W
3	No		E	S
4	No		W	E
...	
142188	No		E	E
142189	No		E	S
142190	No		N	S
142191	No		N	S
142192	No		S	S

	Parameter2_3pm_region	estacion
0	W	invierno
1	W	invierno
2	W	invierno
3	E	invierno
4	N	invierno
...
142188	E	verano
142189	E	verano
142190	N	verano
142191	W	verano
142192	N	verano

[142193 rows x 18 columns]

```
[8]: #Asignamos valores binarios a la variable de fallos y borramos las filas con
      ↪ datos NaN.
df['Failure_today'] = df['Failure_today'].map({'Yes': 1, 'No': 0})
```

```
df.dropna(inplace=True)
df.describe()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 119590 entries, 0 to 142192
```

```
Data columns (total 18 columns):
```

#	Column	Non-Null Count	Dtype
0	Date	119590 non-null	datetime64[ns]
1	Location	119590 non-null	int64
2	Min_Temp	119590 non-null	float64
3	Max_Temp	119590 non-null	float64
4	Parameter1_Speed	119590 non-null	float64
5	Parameter3_9am	119590 non-null	float64
6	Parameter3_3pm	119590 non-null	float64
7	Parameter4_9am	119590 non-null	float64
8	Parameter4_3pm	119590 non-null	float64
9	Parameter5_9am	119590 non-null	float64
10	Parameter5_3pm	119590 non-null	float64
11	Parameter7_9am	119590 non-null	float64
12	Parameter7_3pm	119590 non-null	float64
13	Failure_today	119590 non-null	float64
14	Parameter1_Dir_region	119590 non-null	object
15	Parameter2_9am_region	119590 non-null	object
16	Parameter2_3pm_region	119590 non-null	object
17	estacion	119590 non-null	object

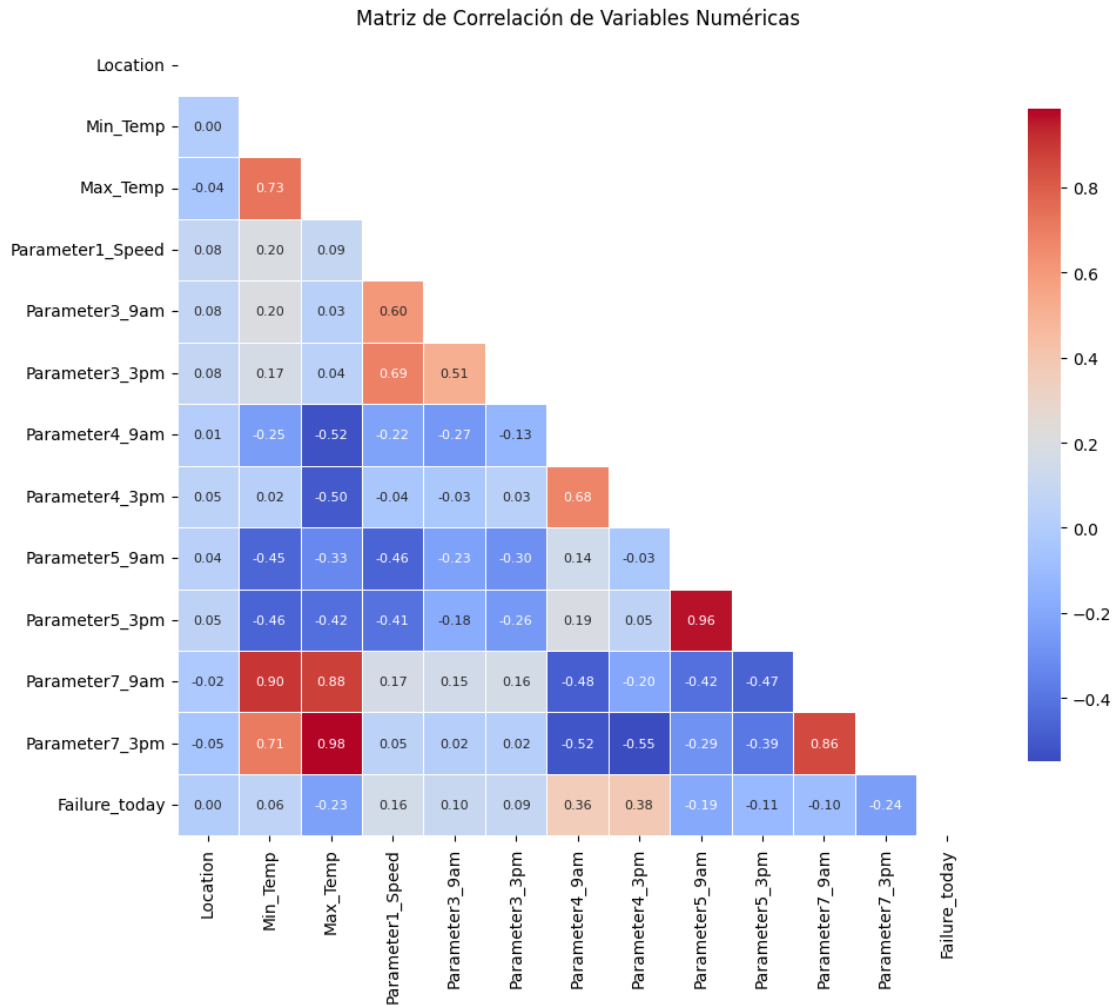
```
dtypes: datetime64[ns](1), float64(12), int64(1), object(4)
```

```
memory usage: 17.3+ MB
```

```
[9]: #Creamos heatmap para observar correlaciones entre variables.
numeric_df = df.select_dtypes(include=['float64', 'int64'])
corr = numeric_df.corr()

mask = np.triu(np.ones_like(corr, dtype=bool))
f, ax = plt.subplots(figsize=(11, 9))
cmap = sns.diverging_palette(230, 20, as_cmap=True)

sns.heatmap(
    corr, annot=True, mask=mask, fmt=".2f", cmap='coolwarm', square=True,
    linewidths=0.5, annot_kws={'size': 8}, cbar_kws={"shrink": .8})
plt.title('Matriz de Correlación de Variables Numéricas')
plt.show()
```



Alta correlación entre Parameter7_9am y Min_Temp, Parameter7_3pm y Max_temp, Parameter5_3pm y Parameter5_9am. Estas dos primeras pueden que también representen temperaturas y por eso presentan tal correlación. Eliminamos algunas para evitar correlación en el modelo.

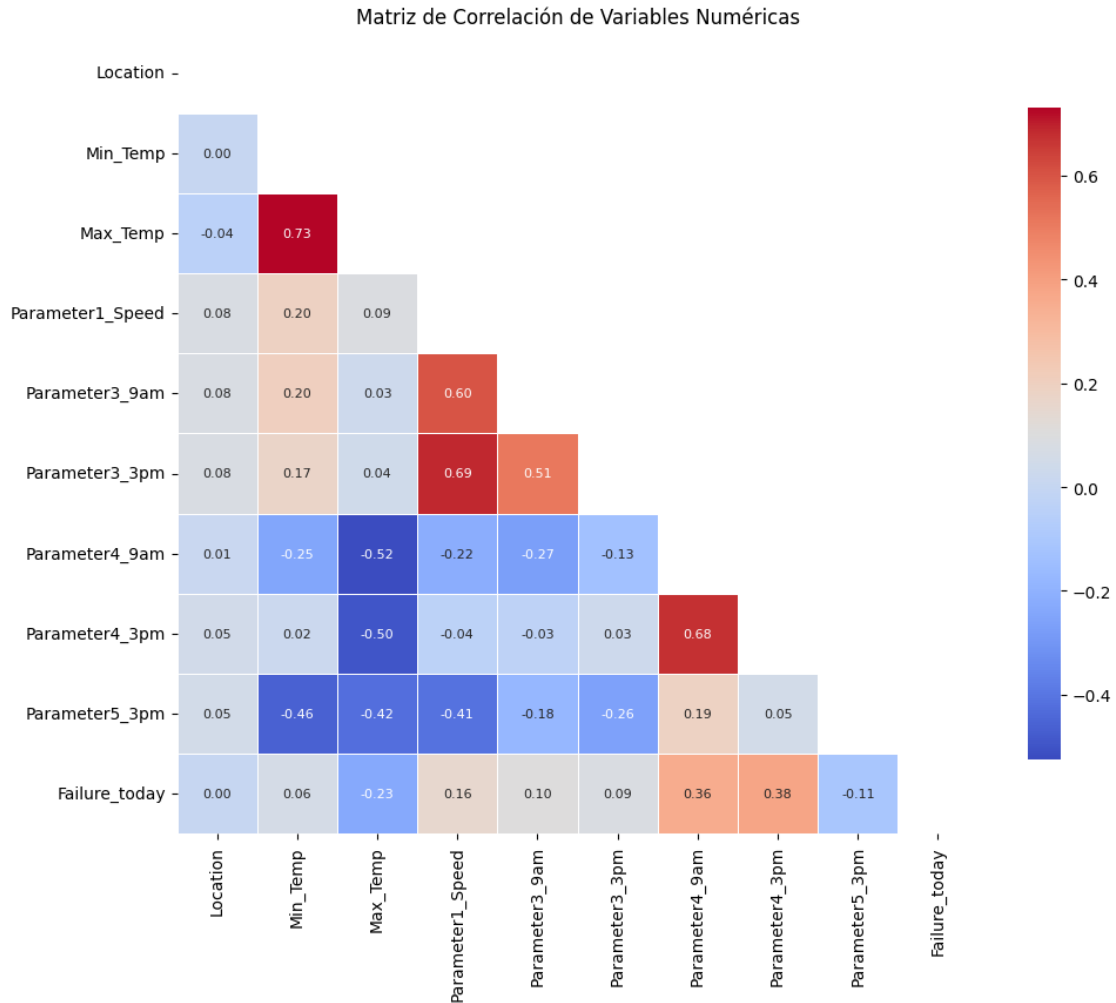
```
[10]: df=df.drop(columns=["Parameter7_9am", "Parameter7_3pm","Parameter5_9am"])
```

```
[11]: #Volvemos a crear el heatmap para observar después del cambio
numeric_df = df.select_dtypes(include=['float64', 'int64'])
corr = numeric_df.corr()

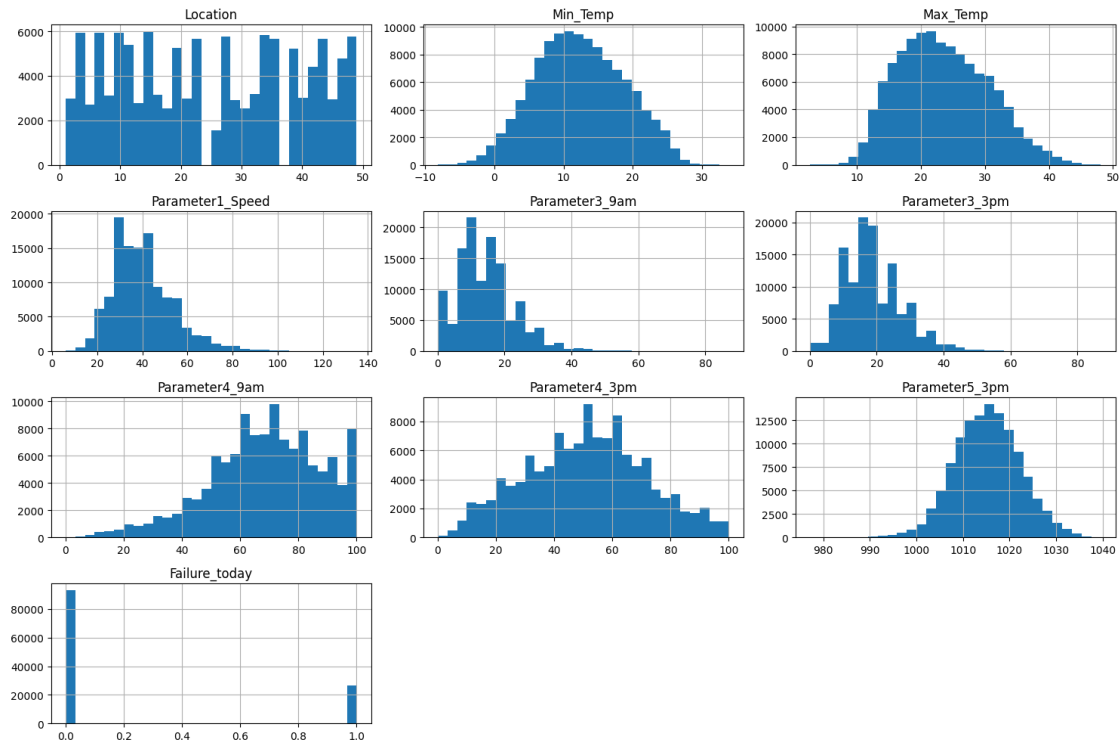
mask = np.triu(np.ones_like(corr, dtype=bool))
f, ax = plt.subplots(figsize=(11, 9))
cmap = sns.diverging_palette(230, 20, as_cmap=True)

sns.heatmap(
```

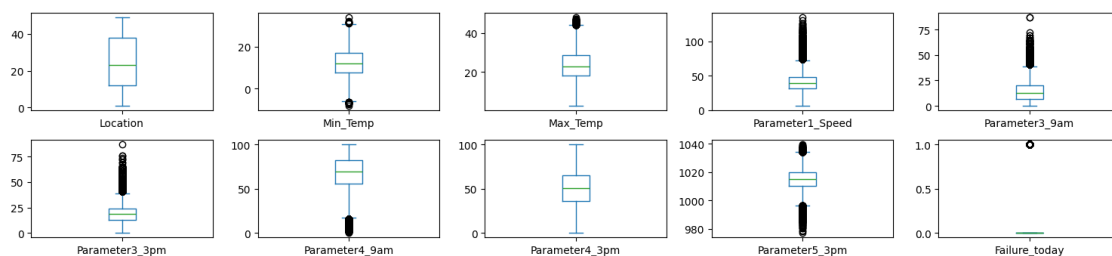
```
corr, annot=True, mask=mask, fmt=".2f", cmap='coolwarm', square=True,
↪linewidths=0.5, annot_kws={'size': 8}, cbar_kws={"shrink": .8})
plt.title('Matriz de Correlación de Variables Numéricas')
plt.show()
```



```
[12]: #Creamos gráficos de barra para observar las distriuciones de nuestras variables
df.select_dtypes(include=['float64', 'int64']).hist(bins=30, figsize=(15, 10))
plt.tight_layout()
plt.show()
```



```
[13]: #Hacemos lo mismo pero con graficos de caja para observar datos extremos
df.select_dtypes(include=['float64', 'int64']).plot(kind='box', subplots=True,
↪ layout=(6, 5), figsize=(15, 10), sharex=False, sharey=False)
plt.tight_layout()
plt.show()
```



2. Ejecute un modelo de probabilidad lineal (MCO) que permita explicar la probabilidad #de que un día se reporte fallo medido por sensor, a partir de las información disponible. #Seleccione las variables dependientes a incluir en el modelo final e interprete su significado

```
[14]: #Transformamos a dummies todas las variables categoricas que utilizaremos en
↪ los modelos (Direcciones de viento, location, estacion) en un dataframe
↪ nuevo.
```

```

df_model = df.drop(columns=["Date"])

#Para direcciones de viento
cols_region = [col for col in df_model.columns if col.endswith('_region')]
df_dummies_region = pd.get_dummies(df_model[cols_region], prefix=cols_region,
    drop_first=True)
df_model = pd.concat([df_model, df_dummies_region], axis=1)
df_model.drop(columns=cols_region, inplace=True)

#Para location
df_model = pd.get_dummies(df_model, columns=['Location'], drop_first=True)

#Para estacion
df_model = pd.get_dummies(df_model, columns=['estacion'], drop_first=True)

#Convertir booleanos a enteros
df_model = df_model.astype({col: int for col in df_model.
    select_dtypes(include='bool').columns})

df_model

```

```

[14]:

```

	Min_Temp	Max_Temp	Parameter1_Speed	Parameter3_9am	Parameter3_3pm	\
0	13.4	22.9	44.0	20.0	24.0	
1	7.4	25.1	44.0	4.0	22.0	
2	12.9	25.7	46.0	19.0	26.0	
3	9.2	28.0	24.0	11.0	9.0	
4	17.5	32.3	41.0	7.0	20.0	
...	
142188	3.5	21.8	31.0	15.0	13.0	
142189	2.8	23.4	31.0	13.0	11.0	
142190	3.6	25.3	22.0	13.0	9.0	
142191	5.4	26.9	37.0	9.0	9.0	
142192	7.8	27.0	28.0	13.0	7.0	

	Parameter4_9am	Parameter4_3pm	Parameter5_3pm	Failure_today	\
0	71.0	22.0	1007.1	0.0	
1	44.0	25.0	1007.8	0.0	
2	38.0	30.0	1008.7	0.0	
3	45.0	16.0	1012.8	0.0	
4	82.0	33.0	1006.0	0.0	
...	
142188	59.0	27.0	1021.2	0.0	
142189	51.0	24.0	1020.3	0.0	
142190	56.0	21.0	1019.1	0.0	
142191	53.0	24.0	1016.8	0.0	
142192	51.0	24.0	1016.5	0.0	

	Parameter1_Dir_region_N	...	Location_43	Location_44	Location_45	\
0	0	...	0	0	0	
1	0	...	0	0	0	
2	0	...	0	0	0	
3	0	...	0	0	0	
4	0	...	0	0	0	
...	
142188	0	...	0	0	0	
142189	0	...	0	0	0	
142190	1	...	0	0	0	
142191	1	...	0	0	0	
142192	0	...	0	0	0	

	Location_46	Location_47	Location_48	Location_49	estacion_otoño	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	
...	
142188	0	0	0	0	0	
142189	0	0	0	0	0	
142190	0	0	0	0	0	
142191	0	0	0	0	0	
142192	0	0	0	0	0	

	estacion_primavera	estacion_verano
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...
142188	0	1
142189	0	1
142190	0	1
142191	0	1
142192	0	1

[119590 rows x 64 columns]

```
[15]: #Definimos nuestro X en base al df creado anteriormente y dropeamos la variable
      ↪ a predecir
X = df_model.drop(columns=["Failure_today"]) # Variables explicativas
X = sm.add_constant(X)

y = df_model['Failure_today'] # Variable dependiente
```

```

modelo = sm.OLS(y, X).fit()
print(modelo.summary())

```

OLS Regression Results

```

=====
Dep. Variable:          Failure_today    R-squared:                0.281
Model:                  OLS              Adj. R-squared:          0.281
Method:                 Least Squares    F-statistic:             741.1
Date:                   Fri, 25 Apr 2025  Prob (F-statistic):      0.00
Time:                   12:57:55          Log-Likelihood:          -44784.
No. Observations:       119590           AIC:                     8.970e+04
Df Residuals:           119526           BIC:                     9.032e+04
Df Model:                63
Covariance Type:        nonrobust
=====

```

```

=====
                                coef    std err          t      P>|t|      [0.025
0.975]
-----
-----
const                6.2809      0.207      30.324      0.000      5.875
6.687
Min_Temp              0.0190      0.000      47.418      0.000      0.018
0.020
Max_Temp             -0.0191      0.000     -47.073      0.000     -0.020
-0.018
Parameter1_Speed      0.0055      0.000      43.040      0.000      0.005
0.006
Parameter3_9am         0.0031      0.000      18.543      0.000      0.003
0.003
Parameter3_3pm        -0.0040      0.000     -22.639      0.000     -0.004
-0.004
Parameter4_9am         0.0077    8.89e-05      86.186      0.000      0.007
0.008
Parameter4_3pm         0.0008    9.84e-05       7.668      0.000      0.001
0.001
Parameter5_3pm        -0.0065      0.000     -32.376      0.000     -0.007
-0.006
Parameter1_Dir_region_N -0.0043      0.004      -1.119      0.263     -0.012
0.003
Parameter1_Dir_region_S  0.0108      0.004       3.011      0.003      0.004
0.018
Parameter1_Dir_region_W  0.0193      0.004       4.959      0.000      0.012
0.027
Parameter2_9am_region_N -0.0093      0.003      -2.818      0.005     -0.016
-0.003
Parameter2_9am_region_S  0.0151      0.003       4.448      0.000      0.008
0.008

```


0.022					
Parameter2_9am_region_W	0.0617	0.004	16.540	0.000	0.054
0.069					
Parameter2_3pm_region_N	0.0078	0.004	2.069	0.039	0.000
0.015					
Parameter2_3pm_region_S	0.0416	0.004	11.817	0.000	0.035
0.048					
Parameter2_3pm_region_W	0.0555	0.004	14.238	0.000	0.048
0.063					
Location_3	-0.0586	0.009	-6.304	0.000	-0.077
-0.040					
Location_4	0.1079	0.009	11.379	0.000	0.089
0.126					
Location_5	-0.0981	0.010	-10.187	0.000	-0.117
-0.079					
Location_6	-0.1816	0.010	-18.754	0.000	-0.201
-0.163					
Location_7	-0.0887	0.009	-9.487	0.000	-0.107
-0.070					
Location_8	-0.0114	0.009	-1.211	0.226	-0.030
0.007					
Location_9	-0.0537	0.010	-5.385	0.000	-0.073
-0.034					
Location_10	-0.0689	0.009	-7.288	0.000	-0.087
-0.050					
Location_11	-0.0249	0.009	-2.662	0.008	-0.043
-0.007					
Location_12	-0.0331	0.010	-3.369	0.001	-0.052
-0.014					
Location_13	-0.1076	0.010	-11.173	0.000	-0.126
-0.089					
Location_14	-0.1001	0.010	-10.255	0.000	-0.119
-0.081					
Location_15	-0.0778	0.010	-8.009	0.000	-0.097
-0.059					
Location_16	-0.1280	0.009	-13.730	0.000	-0.146
-0.110					
Location_17	-0.0675	0.015	-4.401	0.000	-0.098
-0.037					
Location_18	-0.1070	0.011	-9.961	0.000	-0.128
-0.086					
Location_19	-0.1093	0.010	-10.920	0.000	-0.129
-0.090					
Location_20	-0.1485	0.009	-15.695	0.000	-0.167
-0.130					
Location_21	-0.0822	0.009	-8.847	0.000	-0.100
-0.064					
Location_22	-0.0506	0.010	-5.286	0.000	-0.069

-0.032					
Location_23	-0.0695	0.009	-7.353	0.000	-0.088
-0.051					
Location_26	-0.1480	0.011	-13.158	0.000	-0.170
-0.126					
Location_27	-0.1569	0.010	-16.484	0.000	-0.176
-0.138					
Location_28	-0.1539	0.010	-16.182	0.000	-0.173
-0.135					
Location_29	-0.0714	0.009	-7.664	0.000	-0.090
-0.053					
Location_30	-0.0119	0.010	-1.223	0.221	-0.031
0.007					
Location_32	-0.0168	0.009	-1.857	0.063	-0.035
0.001					
Location_33	-0.0215	0.009	-2.311	0.021	-0.040
-0.003					
Location_34	-0.0912	0.009	-9.640	0.000	-0.110
-0.073					
Location_35	-0.0861	0.009	-9.075	0.000	-0.105
-0.067					
Location_36	-0.1753	0.010	-18.142	0.000	-0.194
-0.156					
Location_38	-0.1109	0.010	-11.012	0.000	-0.131
-0.091					
Location_39	-0.0905	0.009	-9.576	0.000	-0.109
-0.072					
Location_40	-0.1033	0.010	-10.456	0.000	-0.123
-0.084					
Location_41	-0.0544	0.009	-5.780	0.000	-0.073
-0.036					
Location_42	0.0747	0.012	6.484	0.000	0.052
0.097					
Location_43	-0.0514	0.009	-5.498	0.000	-0.070
-0.033					
Location_44	-0.0899	0.010	-9.426	0.000	-0.109
-0.071					
Location_45	-0.1332	0.009	-14.295	0.000	-0.151
-0.115					
Location_46	-0.0574	0.010	-5.663	0.000	-0.077
-0.038					
Location_47	-0.0375	0.010	-3.866	0.000	-0.056
-0.018					
Location_48	-0.1797	0.009	-19.017	0.000	-0.198
-0.161					
Location_49	-0.0894	0.009	-9.496	0.000	-0.108
-0.071					
estacion_otoño	0.0299	0.003	8.824	0.000	0.023

```

0.037
estacion_primavera      -0.0151      0.003      -4.625      0.000      -0.022
-0.009
estacion_verano          -0.0175      0.004      -4.011      0.000      -0.026
-0.009
=====
Omnibus:                  9821.479      Durbin-Watson:              1.795
Prob(Omnibus):             0.000      Jarque-Bera (JB):          12305.241
Skew:                      0.779      Prob(JB):                  0.00
Kurtosis:                 2.801      Cond. No.                  2.08e+05
=====

```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 2.08e+05. This might indicate that there are strong multicollinearity or other numerical problems.

R: De acuerdo a los resultados obtenidos del modelo, el aumento de la temperatura minima se asocia con un incremento en la probabilidad de fallo, por el lado contrario una mayor temperatura maxima en el proceso disminuye la probabilidad. También se observa que a mayores velocidades en el Parameter1_Speed también aumenta la probailidad. En cuanto a los parametros, el “Parameter4_9am” es el mas significativo de estos. En cuanto a las “Location” podemos observar que la mayoría de estas representan una disminución en la probabilidad de fallo a excepción de “Location_4” y “Location_42”, esto puede corresponder a que en dichas localizaciones se hace mal uso de la máquina. En las variables de estacion podemos observar que las estaciones de primavera y verano tienden a disminiur la probabilidad de fallo, al contrario de otoño la cual lo aumenta.

3. Ejecute un modelo probit para responder a la pregunta 2.

```

[16]: #Utilizamos la mismas variables dependientes e independientes que usamos en el
      ↪modelo anterior.
modelo = sm.Probit(y, X)
probit = modelo.fit(cov_type="HCO")
print(probit.summary())

mfx= probit.get_margeff()
print(mfx.summary())

```

Optimization terminated successfully.

Current function value: 0.356680

Iterations 7

Probit Regression Results

```

=====
Dep. Variable:      Failure_today      No. Observations:      119590
Model:              Probit              Df Residuals:          119526
Method:              MLE              Df Model:              63
Date:               Fri, 25 Apr 2025      Pseudo R-squ.:          0.3248
Time:               12:57:57              Log-Likelihood:         -42655.

```

converged:	True	LL-Null:	-63172.
Covariance Type:	HCO	LLR p-value:	0.000

	coef	std err	z	P> z	[0.025
0.975]					

const	21.4906	0.951	22.591	0.000	19.626
23.355					
Min_Temp	0.1097	0.002	53.764	0.000	0.106
0.114					
Max_Temp	-0.1299	0.002	-55.838	0.000	-0.134
-0.125					
Parameter1_Speed	0.0211	0.001	35.320	0.000	0.020
0.022					
Parameter3_9am	0.0119	0.001	14.330	0.000	0.010
0.014					
Parameter3_3pm	-0.0135	0.001	-15.865	0.000	-0.015
-0.012					
Parameter4_9am	0.0419	0.001	82.859	0.000	0.041
0.043					
Parameter4_3pm	-0.0027	0.000	-5.963	0.000	-0.004
-0.002					
Parameter5_3pm	-0.0239	0.001	-26.095	0.000	-0.026
-0.022					
Parameter1_Dir_region_N	-0.0444	0.020	-2.264	0.024	-0.083
-0.006					
Parameter1_Dir_region_S	0.0307	0.018	1.735	0.083	-0.004
0.065					
Parameter1_Dir_region_W	0.0887	0.019	4.554	0.000	0.051
0.127					
Parameter2_9am_region_N	-0.0085	0.017	-0.494	0.621	-0.042
0.025					
Parameter2_9am_region_S	0.1290	0.017	7.565	0.000	0.096
0.162					
Parameter2_9am_region_W	0.2710	0.018	15.124	0.000	0.236
0.306					
Parameter2_3pm_region_N	-0.0139	0.019	-0.721	0.471	-0.052
0.024					
Parameter2_3pm_region_S	0.1310	0.017	7.552	0.000	0.097
0.165					
Parameter2_3pm_region_W	0.1588	0.020	8.007	0.000	0.120
0.198					
Location_3	-0.1956	0.046	-4.253	0.000	-0.286
-0.105					
Location_4	0.2484	0.061	4.044	0.000	0.128
0.369					

Location_5 -0.149	-0.2381	0.046	-5.210	0.000	-0.328
Location_6 -0.831	-0.9238	0.047	-19.488	0.000	-1.017
Location_7 -0.291	-0.3808	0.046	-8.267	0.000	-0.471
Location_8 0.403	0.3170	0.044	7.258	0.000	0.231
Location_9 0.151	0.0628	0.045	1.400	0.162	-0.025
Location_10 -0.064	-0.1535	0.046	-3.351	0.001	-0.243
Location_11 -0.024	-0.1270	0.053	-2.416	0.016	-0.230
Location_12 0.155	0.0671	0.045	1.492	0.136	-0.021
Location_13 -0.421	-0.5069	0.044	-11.522	0.000	-0.593
Location_14 0.014	-0.0776	0.047	-1.668	0.095	-0.169
Location_15 0.025	-0.0631	0.045	-1.399	0.162	-0.151
Location_16 -0.271	-0.3587	0.045	-8.054	0.000	-0.446
Location_17 0.183	0.0277	0.079	0.350	0.726	-0.127
Location_18 -0.248	-0.3442	0.049	-6.981	0.000	-0.441
Location_19 -0.201	-0.2927	0.047	-6.281	0.000	-0.384
Location_20 -0.463	-0.5514	0.045	-12.193	0.000	-0.640
Location_21 -0.465	-0.5649	0.051	-11.114	0.000	-0.664
Location_22 0.086	-0.0144	0.051	-0.282	0.778	-0.114
Location_23 -0.229	-0.3140	0.044	-7.204	0.000	-0.399
Location_26 -0.683	-0.7974	0.058	-13.641	0.000	-0.912
Location_27 -0.436	-0.5230	0.044	-11.838	0.000	-0.610
Location_28 -0.422	-0.5060	0.043	-11.833	0.000	-0.590
Location_29 -0.398	-0.4938	0.049	-10.054	0.000	-0.590
Location_30 0.212	0.1161	0.049	2.364	0.018	0.020

Location_32 0.172	0.0874	0.043	2.025	0.043	0.003
Location_33 0.185	0.0961	0.046	2.108	0.035	0.007
Location_34 -0.341	-0.4255	0.043	-9.915	0.000	-0.510
Location_35 -0.129	-0.2180	0.045	-4.820	0.000	-0.307
Location_36 -0.531	-0.6216	0.046	-13.486	0.000	-0.712
Location_38 -0.163	-0.2526	0.046	-5.522	0.000	-0.342
Location_39 -0.125	-0.2139	0.045	-4.707	0.000	-0.303
Location_40 -0.038	-0.1297	0.047	-2.764	0.006	-0.222
Location_41 0.004	-0.0849	0.045	-1.871	0.061	-0.174
Location_42 0.361	0.2088	0.077	2.698	0.007	0.057
Location_43 -0.081	-0.1769	0.049	-3.632	0.000	-0.272
Location_44 -0.211	-0.2959	0.043	-6.860	0.000	-0.380
Location_45 -0.442	-0.5283	0.044	-12.006	0.000	-0.615
Location_46 0.064	-0.0289	0.047	-0.611	0.541	-0.121
Location_47 0.042	-0.0459	0.045	-1.028	0.304	-0.134
Location_48 -0.519	-0.6068	0.045	-13.571	0.000	-0.694
Location_49 -0.606	-0.7232	0.060	-12.119	0.000	-0.840
estacion_otoño 0.082	0.0477	0.017	2.727	0.006	0.013
estacion_primavera -0.122	-0.1530	0.016	-9.545	0.000	-0.184
estacion_verano -0.235	-0.2773	0.022	-12.865	0.000	-0.319

=====

=====

Probit Marginal Effects

=====

Dep. Variable:	Failure_today
Method:	dydx
At:	overall

=====

=====	dy/dx	std err	z	P> z	[0.025
0.975]					

Min_Temp	0.0220	0.000	55.585	0.000	0.021
0.023					
Max_Temp	-0.0260	0.000	-59.057	0.000	-0.027
-0.025					
Parameter1_Speed	0.0042	0.000	36.045	0.000	0.004
0.004					
Parameter3_9am	0.0024	0.000	14.378	0.000	0.002
0.003					
Parameter3_3pm	-0.0027	0.000	-15.925	0.000	-0.003
-0.002					
Parameter4_9am	0.0084	8.78e-05	95.524	0.000	0.008
0.009					
Parameter4_3pm	-0.0005	9.14e-05	-5.971	0.000	-0.001
-0.000					
Parameter5_3pm	-0.0048	0.000	-26.346	0.000	-0.005
-0.004					
Parameter1_Dir_region_N	-0.0089	0.004	-2.265	0.024	-0.017
-0.001					
Parameter1_Dir_region_S	0.0062	0.004	1.735	0.083	-0.001
0.013					
Parameter1_Dir_region_W	0.0178	0.004	4.552	0.000	0.010
0.025					
Parameter2_9am_region_N	-0.0017	0.003	-0.494	0.621	-0.008
0.005					
Parameter2_9am_region_S	0.0258	0.003	7.568	0.000	0.019
0.033					
Parameter2_9am_region_W	0.0543	0.004	15.162	0.000	0.047
0.061					
Parameter2_3pm_region_N	-0.0028	0.004	-0.721	0.471	-0.010
0.005					
Parameter2_3pm_region_S	0.0262	0.003	7.558	0.000	0.019
0.033					
Parameter2_3pm_region_W	0.0318	0.004	8.012	0.000	0.024
0.040					
Location_3	-0.0392	0.009	-4.257	0.000	-0.057
-0.021					
Location_4	0.0497	0.012	4.044	0.000	0.026
0.074					
Location_5	-0.0477	0.009	-5.213	0.000	-0.066
-0.030					
Location_6	-0.1849	0.009	-19.697	0.000	-0.203
-0.167					
Location_7	-0.0762	0.009	-8.284	0.000	-0.094

-0.058					
Location_8	0.0635	0.009	7.265	0.000	0.046
0.081					
Location_9	0.0126	0.009	1.400	0.162	-0.005
0.030					
Location_10	-0.0307	0.009	-3.353	0.001	-0.049
-0.013					
Location_11	-0.0254	0.011	-2.418	0.016	-0.046
-0.005					
Location_12	0.0134	0.009	1.492	0.136	-0.004
0.031					
Location_13	-0.1015	0.009	-11.561	0.000	-0.119
-0.084					
Location_14	-0.0155	0.009	-1.668	0.095	-0.034
0.003					
Location_15	-0.0126	0.009	-1.399	0.162	-0.030
0.005					
Location_16	-0.0718	0.009	-8.074	0.000	-0.089
-0.054					
Location_17	0.0055	0.016	0.350	0.726	-0.025
0.037					
Location_18	-0.0689	0.010	-6.991	0.000	-0.088
-0.050					
Location_19	-0.0586	0.009	-6.289	0.000	-0.077
-0.040					
Location_20	-0.1104	0.009	-12.239	0.000	-0.128
-0.093					
Location_21	-0.1131	0.010	-11.147	0.000	-0.133
-0.093					
Location_22	-0.0029	0.010	-0.282	0.778	-0.023
0.017					
Location_23	-0.0629	0.009	-7.216	0.000	-0.080
-0.046					
Location_26	-0.1596	0.012	-13.697	0.000	-0.182
-0.137					
Location_27	-0.1047	0.009	-11.869	0.000	-0.122
-0.087					
Location_28	-0.1013	0.009	-11.864	0.000	-0.118
-0.085					
Location_29	-0.0989	0.010	-10.091	0.000	-0.118
-0.080					
Location_30	0.0232	0.010	2.364	0.018	0.004
0.042					
Location_32	0.0175	0.009	2.025	0.043	0.001
0.034					
Location_33	0.0192	0.009	2.109	0.035	0.001
0.037					
Location_34	-0.0852	0.009	-9.941	0.000	-0.102

-0.068					
Location_35	-0.0436	0.009	-4.823	0.000	-0.061
-0.026					
Location_36	-0.1244	0.009	-13.555	0.000	-0.142
-0.106					
Location_38	-0.0506	0.009	-5.526	0.000	-0.069
-0.033					
Location_39	-0.0428	0.009	-4.709	0.000	-0.061
-0.025					
Location_40	-0.0260	0.009	-2.763	0.006	-0.044
-0.008					
Location_41	-0.0170	0.009	-1.872	0.061	-0.035
0.001					
Location_42	0.0418	0.015	2.698	0.007	0.011
0.072					
Location_43	-0.0354	0.010	-3.635	0.000	-0.054
-0.016					
Location_44	-0.0592	0.009	-6.868	0.000	-0.076
-0.042					
Location_45	-0.1058	0.009	-12.054	0.000	-0.123
-0.089					
Location_46	-0.0058	0.009	-0.611	0.541	-0.024
0.013					
Location_47	-0.0092	0.009	-1.028	0.304	-0.027
0.008					
Location_48	-0.1215	0.009	-13.616	0.000	-0.139
-0.104					
Location_49	-0.1448	0.012	-12.178	0.000	-0.168
-0.121					
estacion_otoño	0.0096	0.004	2.727	0.006	0.003
0.016					
estacion_primavera	-0.0306	0.003	-9.548	0.000	-0.037
-0.024					
estacion_verano	-0.0555	0.004	-12.898	0.000	-0.064
-0.047					
=====					
=====					

R: Interpretamos similarmente a la pregunta anterior, en base a estos resultados las temperaturas mínimas y máximas se comportan de la misma forma que en el modelo (MCO), Min_Temp tiene $dy/dx = 0.0220$ y Max_Temp tiene $dy/dx = -0.0260$. En cuanto a las velocidades del viento “Parameter1_Speed” se comporta igual que antes, indicando un aumento de probabilidad cuando este aumenta ($dy/dx = 0.0042$). En el caso de las estaciones tampoco cambia el comportamiento. En cuanto a las “Location” se observan cambios con respecto al modelo anterior, indicando que algunas location ahora aumentan la probabilidad de fallo cuando antes indicaban que no.

```
[17]: modelo_logit = sm.Logit(y, X).fit(displ=False)
      print(modelo_logit.summary())
```

```
mfx= modelo_logit.get_margeff()
print(mfx.summary())
```

Logit Regression Results

```
=====
Dep. Variable:          Failure_today    No. Observations:          119590
Model:                  Logit           Df Residuals:            119526
Method:                 MLE            Df Model:                63
Date:                  Fri, 25 Apr 2025   Pseudo R-squ.:            0.3272
Time:                  12:58:06          Log-Likelihood:           -42502.
converged:              True             LL-Null:                  -63172.
Covariance Type:        nonrobust         LLR p-value:              0.000
=====
```

```
=====
                                coef      std err          z      P>|z|      [0.025
0.975]
-----
-----
const                        36.9592      1.666      22.187      0.000      33.694
40.224
Min_Temp                     0.1967      0.004      52.421      0.000      0.189
0.204
Max_Temp                    -0.2396      0.004     -59.421      0.000     -0.248
-0.232
Parameter1_Speed             0.0371      0.001      35.677      0.000      0.035
0.039
Parameter3_9am               0.0203      0.001      13.876      0.000      0.017
0.023
Parameter3_3pm              -0.0227      0.001     -15.237      0.000     -0.026
-0.020
Parameter4_9am               0.0761      0.001      86.470      0.000      0.074
0.078
Parameter4_3pm              -0.0058      0.001      -7.258      0.000     -0.007
-0.004
Parameter5_3pm              -0.0412      0.002     -25.622      0.000     -0.044
-0.038
Parameter1_Dir_region_N     -0.0955      0.036      -2.688      0.007     -0.165
-0.026
Parameter1_Dir_region_S      0.0347      0.032       1.079      0.281     -0.028
0.098
Parameter1_Dir_region_W      0.1410      0.035       4.056      0.000      0.073
0.209
Parameter2_9am_region_N     -0.0212      0.031      -0.682      0.495     -0.082
0.040
Parameter2_9am_region_S      0.2253      0.031       7.204      0.000      0.164
0.287
Parameter2_9am_region_W      0.4770      0.032      14.752      0.000      0.414
0.540
=====
```

0.540					
Parameter2_3pm_region_N	-0.0327	0.035	-0.941	0.347	-0.101
0.035					
Parameter2_3pm_region_S	0.2192	0.031	7.021	0.000	0.158
0.280					
Parameter2_3pm_region_W	0.2618	0.035	7.493	0.000	0.193
0.330					
Location_3	-0.4099	0.082	-4.977	0.000	-0.571
-0.248					
Location_4	0.3676	0.112	3.294	0.001	0.149
0.586					
Location_5	-0.4069	0.083	-4.888	0.000	-0.570
-0.244					
Location_6	-1.7276	0.083	-20.868	0.000	-1.890
-1.565					
Location_7	-0.7195	0.084	-8.607	0.000	-0.883
-0.556					
Location_8	0.6299	0.080	7.907	0.000	0.474
0.786					
Location_9	0.2190	0.082	2.668	0.008	0.058
0.380					
Location_10	-0.3012	0.083	-3.629	0.000	-0.464
-0.139					
Location_11	-0.3160	0.094	-3.375	0.001	-0.499
-0.133					
Location_12	0.1613	0.081	1.981	0.048	0.002
0.321					
Location_13	-0.9266	0.079	-11.752	0.000	-1.081
-0.772					
Location_14	-0.0201	0.084	-0.239	0.811	-0.185
0.144					
Location_15	-0.0402	0.082	-0.488	0.625	-0.201
0.121					
Location_16	-0.6818	0.078	-8.686	0.000	-0.836
-0.528					
Location_17	0.1855	0.143	1.294	0.196	-0.096
0.467					
Location_18	-0.6223	0.089	-6.992	0.000	-0.797
-0.448					
Location_19	-0.5307	0.082	-6.479	0.000	-0.691
-0.370					
Location_20	-1.0022	0.080	-12.472	0.000	-1.160
-0.845					
Location_21	-1.0492	0.092	-11.358	0.000	-1.230
-0.868					
Location_22	-0.0334	0.093	-0.358	0.720	-0.216
0.149					
Location_23	-0.5814	0.078	-7.435	0.000	-0.735

-0.428					
Location_26	-1.4462	0.104	-13.849	0.000	-1.651
-1.241					
Location_27	-0.9191	0.079	-11.673	0.000	-1.073
-0.765					
Location_28	-0.8777	0.077	-11.470	0.000	-1.028
-0.728					
Location_29	-0.9575	0.086	-11.113	0.000	-1.126
-0.789					
Location_30	0.2137	0.089	2.409	0.016	0.040
0.388					
Location_32	0.1934	0.080	2.416	0.016	0.037
0.350					
Location_33	0.1964	0.085	2.324	0.020	0.031
0.362					
Location_34	-0.7737	0.076	-10.183	0.000	-0.923
-0.625					
Location_35	-0.3658	0.083	-4.417	0.000	-0.528
-0.203					
Location_36	-1.1410	0.082	-13.978	0.000	-1.301
-0.981					
Location_38	-0.4240	0.082	-5.171	0.000	-0.585
-0.263					
Location_39	-0.3786	0.080	-4.719	0.000	-0.536
-0.221					
Location_40	-0.1056	0.088	-1.207	0.228	-0.277
0.066					
Location_41	-0.1758	0.082	-2.137	0.033	-0.337
-0.015					
Location_42	0.3162	0.142	2.224	0.026	0.037
0.595					
Location_43	-0.3984	0.086	-4.623	0.000	-0.567
-0.230					
Location_44	-0.5270	0.077	-6.841	0.000	-0.678
-0.376					
Location_45	-0.9763	0.079	-12.426	0.000	-1.130
-0.822					
Location_46	-0.0443	0.084	-0.528	0.597	-0.208
0.120					
Location_47	-0.0836	0.080	-1.048	0.295	-0.240
0.073					
Location_48	-1.0698	0.079	-13.492	0.000	-1.225
-0.914					
Location_49	-1.3775	0.106	-13.019	0.000	-1.585
-1.170					
estacion_otoño	0.0667	0.030	2.211	0.027	0.008
0.126					
estacion_primavera	-0.2541	0.028	-9.047	0.000	-0.309

-0.199					
estacion_verano	-0.4959	0.038	-13.218	0.000	-0.569
-0.422					

=====

=====

Logit Marginal Effects

Dep. Variable: Failure_today

Method: dydx

At: overall

=====

=====

	dy/dx	std err	z	P> z	[0.025
--	-------	---------	---	------	--------

0.975]

Min_Temp	0.0221	0.000	54.507	0.000	0.021
0.023					
Max_Temp	-0.0269	0.000	-62.449	0.000	-0.028
-0.026					
Parameter1_Speed	0.0042	0.000	36.480	0.000	0.004
0.004					
Parameter3_9am	0.0023	0.000	13.925	0.000	0.002
0.003					
Parameter3_3pm	-0.0026	0.000	-15.305	0.000	-0.003
-0.002					
Parameter4_9am	0.0086	8.72e-05	98.095	0.000	0.008
0.009					
Parameter4_3pm	-0.0007	8.98e-05	-7.262	0.000	-0.001
-0.000					
Parameter5_3pm	-0.0046	0.000	-25.913	0.000	-0.005
-0.004					
Parameter1_Dir_region_N	-0.0107	0.004	-2.688	0.007	-0.019
-0.003					
Parameter1_Dir_region_S	0.0039	0.004	1.079	0.281	-0.003
0.011					
Parameter1_Dir_region_W	0.0158	0.004	4.057	0.000	0.008
0.024					
Parameter2_9am_region_N	-0.0024	0.003	-0.682	0.495	-0.009
0.004					
Parameter2_9am_region_S	0.0253	0.004	7.208	0.000	0.018
0.032					
Parameter2_9am_region_W	0.0536	0.004	14.799	0.000	0.047
0.061					
Parameter2_3pm_region_N	-0.0037	0.004	-0.941	0.347	-0.011
0.004					
Parameter2_3pm_region_S	0.0246	0.004	7.028	0.000	0.018
0.031					

Parameter2_3pm_region_W 0.037	0.0294	0.004	7.501	0.000	0.022
Location_3 -0.028	-0.0461	0.009	-4.978	0.000	-0.064
Location_4 0.066	0.0413	0.013	3.295	0.001	0.017
Location_5 -0.027	-0.0457	0.009	-4.890	0.000	-0.064
Location_6 -0.176	-0.1941	0.009	-21.013	0.000	-0.212
Location_7 -0.062	-0.0809	0.009	-8.617	0.000	-0.099
Location_8 0.088	0.0708	0.009	7.911	0.000	0.053
Location_9 0.043	0.0246	0.009	2.668	0.008	0.007
Location_10 -0.016	-0.0338	0.009	-3.630	0.000	-0.052
Location_11 -0.015	-0.0355	0.011	-3.376	0.001	-0.056
Location_12 0.036	0.0181	0.009	1.981	0.048	0.000
Location_13 -0.087	-0.1041	0.009	-11.778	0.000	-0.121
Location_14 0.016	-0.0023	0.009	-0.239	0.811	-0.021
Location_15 0.014	-0.0045	0.009	-0.488	0.625	-0.023
Location_16 -0.059	-0.0766	0.009	-8.698	0.000	-0.094
Location_17 0.052	0.0208	0.016	1.294	0.196	-0.011
Location_18 -0.050	-0.0699	0.010	-6.999	0.000	-0.090
Location_19 -0.042	-0.0596	0.009	-6.485	0.000	-0.078
Location_20 -0.095	-0.1126	0.009	-12.506	0.000	-0.130
Location_21 -0.098	-0.1179	0.010	-11.381	0.000	-0.138
Location_22 0.017	-0.0038	0.010	-0.358	0.720	-0.024
Location_23 -0.048	-0.0653	0.009	-7.442	0.000	-0.083
Location_26 -0.140	-0.1625	0.012	-13.893	0.000	-0.185
Location_27 -0.086	-0.1033	0.009	-11.701	0.000	-0.121

Location_28	-0.0986	0.009	-11.499	0.000	-0.115
-0.082					
Location_29	-0.1076	0.010	-11.133	0.000	-0.127
-0.089					
Location_30	0.0240	0.010	2.409	0.016	0.004
0.044					
Location_32	0.0217	0.009	2.416	0.016	0.004
0.039					
Location_33	0.0221	0.009	2.324	0.020	0.003
0.041					
Location_34	-0.0869	0.009	-10.201	0.000	-0.104
-0.070					
Location_35	-0.0411	0.009	-4.419	0.000	-0.059
-0.023					
Location_36	-0.1282	0.009	-14.027	0.000	-0.146
-0.110					
Location_38	-0.0476	0.009	-5.175	0.000	-0.066
-0.030					
Location_39	-0.0425	0.009	-4.721	0.000	-0.060
-0.025					
Location_40	-0.0119	0.010	-1.207	0.228	-0.031
0.007					
Location_41	-0.0198	0.009	-2.137	0.033	-0.038
-0.002					
Location_42	0.0355	0.016	2.224	0.026	0.004
0.067					
Location_43	-0.0448	0.010	-4.625	0.000	-0.064
-0.026					
Location_44	-0.0592	0.009	-6.847	0.000	-0.076
-0.042					
Location_45	-0.1097	0.009	-12.459	0.000	-0.127
-0.092					
Location_46	-0.0050	0.009	-0.528	0.597	-0.023
0.013					
Location_47	-0.0094	0.009	-1.048	0.295	-0.027
0.008					
Location_48	-0.1202	0.009	-13.536	0.000	-0.138
-0.103					
Location_49	-0.1548	0.012	-13.051	0.000	-0.178
-0.132					
estacion_otoño	0.0075	0.003	2.211	0.027	0.001
0.014					
estacion_primavera	-0.0286	0.003	-9.057	0.000	-0.035
-0.022					
estacion_verano	-0.0557	0.004	-13.249	0.000	-0.064
-0.047					

=====

=====

R: En este caso los resultados del modelo logit obtenemos comportamientos muy similares a los obtenidos en el modelo probit. Las temperaturas se comportan de la misma manera pero con una leve diferencia en su magnitud. El parametro de velocidad “Parameter1_Speed” se comporta de igual forma hasta en su magnitud ($dy/dx = 0.0042$). En cuanto a las location tampoco se observan mayores diferencias, asda asdasdsa. Las estaciones se comportan de igual manera.

5. Comente los resultados obtenidos en 2, 3 y 4. ¿Cuáles y por qué existen las diferencias entre los resultados?. En su opinión, ¿Cuál sería el más adecuado para responder la pregunta de investigación y por qué? ¿Qué variables resultaron ser robustas a la especificación?

R: De acuerdo a los resultados obtenidos en los 3 modelos, podemos observar que las variables a estudiar se comportan generalmente de igual forma, en el sentido de si aumentan o disminuyen la probabilidad de que ocurra un fallo. Una diferencia entre los modelos ocurre entre el modelo OLS y los modelos Probit y Logit con respecto a la variable de “Location”, según el modelo OLS casi todas las “Location” indicaban una disminución en la probabilidad de fallo, pero tanto en el modelo Probit como Logit se observa que varias de estas invierten sus comportamientos. En mi opinión de acuerdo a que modelo es más adecuado para nuestro caso de estudio, el modelo OLS (MCO) no es adecuado dado que no se ajusta bien a la variable binaria a predecir (Failure_Today), en cambio los modelos Probit y Logit ya que están diseñados para manejar variables dependientes binarias. En los dos modelos se obtuvo resultados bastante similares en cuanto a sus valores de R y en cuanto a sus coeficientes marginales por lo que entre elegir uno u otro no hay mayor diferencia, sin embargo el modelo Probit puede ser más adecuado para nuestros datos dado que no tenemos tantos datos extremos.

6. Agregue la data a nivel mensual, usando la data promedio de las variables (ignorando aquellas categoricas, como la direccion del viento). En particular, genere una variable que cuente la cantidad de fallos observados en un mes, utilice un valor de 0 si en ese mes no se reporto fallos en ningun dia. Use un modelo Poisson para explicar el numero de fallas por mes. Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.

```
[18]: #Creamos un nuevo df para el modelo poisson y excluimos las variables
      ↪categoricas
df_poisson = df.copy()

df_poisson = df_poisson.select_dtypes(exclude=['object', 'category'])

df_poisson
```

```
[18]:
```

	Date	Location	Min_Temp	Max_Temp	Parameter1_Speed	\
0	2008-12-01	3	13.4	22.9	44.0	
1	2008-12-02	3	7.4	25.1	44.0	
2	2008-12-03	3	12.9	25.7	46.0	
3	2008-12-04	3	9.2	28.0	24.0	
4	2008-12-05	3	17.5	32.3	41.0	
...	
142188	2017-06-20	42	3.5	21.8	31.0	
142189	2017-06-21	42	2.8	23.4	31.0	
142190	2017-06-22	42	3.6	25.3	22.0	
142191	2017-06-23	42	5.4	26.9	37.0	

142192	2017-06-24	42	7.8	27.0	28.0
--------	------------	----	-----	------	------

	Parameter3_9am	Parameter3_3pm	Parameter4_9am	Parameter4_3pm	\
0	20.0	24.0	71.0	22.0	
1	4.0	22.0	44.0	25.0	
2	19.0	26.0	38.0	30.0	
3	11.0	9.0	45.0	16.0	
4	7.0	20.0	82.0	33.0	
...	
142188	15.0	13.0	59.0	27.0	
142189	13.0	11.0	51.0	24.0	
142190	13.0	9.0	56.0	21.0	
142191	9.0	9.0	53.0	24.0	
142192	13.0	7.0	51.0	24.0	

	Parameter5_3pm	Failure_today
0	1007.1	0.0
1	1007.8	0.0
2	1008.7	0.0
3	1012.8	0.0
4	1006.0	0.0
...
142188	1021.2	0.0
142189	1020.3	0.0
142190	1019.1	0.0
142191	1016.8	0.0
142192	1016.5	0.0

[119590 rows x 11 columns]

```
[19]: df_poisson=df_poisson[df_poisson["Date"].dt.year>2008] #Consideramos desde el
      ↪ año 2009 en adelante porque faltan algunos location
df_poisson["Mes"] = df_poisson["Date"].dt.to_period("M")

df_poisson
```

C:\Users\Nacho\AppData\Local\Temp\ipykernel_6444\1462064157.py:2:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_poisson["Mes"] = df_poisson["Date"].dt.to_period("M")
```

```
[19]:      Date Location Min_Temp Max_Temp Parameter1_Speed \
30    2009-01-01      3    11.3    26.5          56.0
31    2009-01-02      3     9.6    23.9          41.0
```

32	2009-01-03	3	10.5	28.8	26.0
33	2009-01-04	3	12.3	34.6	37.0
34	2009-01-05	3	12.9	35.8	41.0
...
142188	2017-06-20	42	3.5	21.8	31.0
142189	2017-06-21	42	2.8	23.4	31.0
142190	2017-06-22	42	3.6	25.3	22.0
142191	2017-06-23	42	5.4	26.9	37.0
142192	2017-06-24	42	7.8	27.0	28.0

	Parameter3_9am	Parameter3_3pm	Parameter4_9am	Parameter4_3pm	\
30	19.0	31.0	46.0	26.0	
31	19.0	11.0	44.0	22.0	
32	11.0	7.0	43.0	22.0	
33	6.0	17.0	41.0	12.0	
34	6.0	26.0	41.0	9.0	
...	
142188	15.0	13.0	59.0	27.0	
142189	13.0	11.0	51.0	24.0	
142190	13.0	9.0	56.0	21.0	
142191	9.0	9.0	53.0	24.0	
142192	13.0	7.0	51.0	24.0	

	Parameter5_3pm	Failure_today	Mes
30	1003.2	0.0	2009-01
31	1013.1	0.0	2009-01
32	1014.8	0.0	2009-01
33	1010.3	0.0	2009-01
34	1009.2	0.0	2009-01
...
142188	1021.2	0.0	2017-06
142189	1020.3	0.0	2017-06
142190	1019.1	0.0	2017-06
142191	1016.8	0.0	2017-06
142192	1016.5	0.0	2017-06

[117793 rows x 12 columns]

```
[20]: promedios = df_poisson.groupby(["Mes", "Location"]).agg({"Min_Temp": "mean",
    "Max_Temp": "mean", "Parameter1_Speed": "mean", "Parameter3_9am": "mean",
    "Parameter3_3pm": "mean", "Parameter4_9am": "mean", "Parameter4_3pm":
    ↪ "mean", "Parameter5_3pm": "mean",
    "Failure_today": "sum"}).reset_index()
promedios
```

```
[20]:
```

	Mes	Location	Min_Temp	Max_Temp	Parameter1_Speed	\
0	2009-01	1	17.975862	31.868966	39.965517	

1	2009-01	3	16.312903	34.658065	42.677419
2	2009-01	4	22.422581	36.058065	51.258065
3	2009-01	5	16.250000	32.733333	41.300000
4	2009-01	6	10.617241	28.548276	48.620690
...
4071	2017-06	45	4.424000	14.744000	24.040000
4072	2017-06	46	10.100000	18.356000	34.120000
4073	2017-06	47	8.736000	18.616000	34.000000
4074	2017-06	48	11.788889	17.816667	37.166667
4075	2017-06	49	5.800000	18.754167	27.666667

	Parameter3_9am	Parameter3_3pm	Parameter4_9am	Parameter4_3pm \
0	10.448276	17.931034	38.689655	23.827586
1	11.935484	18.548387	41.903226	17.870968
2	18.516129	25.032258	37.096774	24.516129
3	7.300000	17.466667	65.466667	35.933333
4	20.172414	22.241379	50.586207	24.379310
...
4071	4.960000	9.280000	97.840000	67.760000
4072	16.440000	16.440000	87.200000	70.880000
4073	9.520000	16.320000	88.520000	67.280000
4074	14.666667	19.000000	72.166667	68.666667
4075	11.375000	12.833333	66.041667	35.875000

	Parameter5_3pm	Failure_today
0	1012.324138	0.0
1	1009.770968	1.0
2	1004.732258	3.0
3	1012.353333	3.0
4	1011.451724	0.0
...
4071	1026.476000	3.0
4072	1023.492000	13.0
4073	1022.168000	9.0
4074	1024.283333	4.0
4075	1027.033333	0.0

[4076 rows x 11 columns]

```
[21]: poisson= smf.poisson("Failure_today ~ C(Location) + Min_Temp + Max_Temp +
↪Parameter1_Speed + Parameter3_9am + Parameter3_3pm + Parameter4_9am +
↪Parameter4_3pm + Parameter5_3pm",
                        data=promedios).fit()

print(poisson.summary())
```

Optimization terminated successfully.

Current function value: 2.227737

Iterations 8

Poisson Regression Results

```

=====
Dep. Variable:      Failure_today    No. Observations:      4076
Model:              Poisson          Df Residuals:          4024
Method:             MLE              Df Model:              51
Date:              Fri, 25 Apr 2025   Pseudo R-squ.:         0.3207
Time:              12:58:15          Log-Likelihood:        -9080.3
converged:          True              LL-Null:               -13366.
Covariance Type:    nonrobust         LLR p-value:           0.000
=====

```

```

=====
                                coef      std err          z      P>|z|      [0.025
0.975]
-----
-----
Intercept                21.1181      2.908      7.262      0.000      15.418
26.818
C(Location) [T.3]         0.0610      0.066      0.931      0.352      -0.067
0.189
C(Location) [T.4]         0.0868      0.082      1.063      0.288      -0.073
0.247
C(Location) [T.5]        -0.1696      0.068     -2.497      0.013      -0.303
-0.036
C(Location) [T.6]        -0.3748      0.074     -5.056      0.000      -0.520
-0.229
C(Location) [T.7]        -0.1099      0.067     -1.650      0.099      -0.240
0.021
C(Location) [T.8]        -0.0205      0.060     -0.342      0.732      -0.138
0.097
C(Location) [T.9]        -0.0363      0.063     -0.572      0.567      -0.161
0.088
C(Location) [T.10]       -0.0183      0.073     -0.250      0.802      -0.162
0.125
C(Location) [T.11]       -0.0322      0.070     -0.459      0.646      -0.169
0.105
C(Location) [T.12]        0.0211      0.063      0.337      0.736      -0.101
0.144
C(Location) [T.13]       -0.2443      0.066     -3.712      0.000      -0.373
-0.115
C(Location) [T.14]       -0.3463      0.064     -5.405      0.000      -0.472
-0.221
C(Location) [T.15]       -0.0981      0.069     -1.420      0.156      -0.233
0.037
C(Location) [T.16]       -0.6157      0.059    -10.487      0.000      -0.731
-0.501
C(Location) [T.17]       -0.5770      0.111     -5.203      0.000      -0.794
-0.360

```

C(Location) [T.18] -0.167	-0.3062	0.071	-4.326	0.000	-0.445
C(Location) [T.19] -0.224	-0.3500	0.064	-5.433	0.000	-0.476
C(Location) [T.20] -0.123	-0.2537	0.067	-3.807	0.000	-0.384
C(Location) [T.21] 0.036	-0.1130	0.076	-1.482	0.138	-0.262
C(Location) [T.22] 0.110	-0.0414	0.077	-0.535	0.593	-0.193
C(Location) [T.23] 0.135	0.0094	0.064	0.146	0.884	-0.117
C(Location) [T.26] -0.083	-0.2527	0.086	-2.925	0.003	-0.422
C(Location) [T.27] -0.514	-0.6315	0.060	-10.511	0.000	-0.749
C(Location) [T.28] -0.472	-0.5966	0.064	-9.377	0.000	-0.721
C(Location) [T.29] 0.022	-0.1018	0.063	-1.608	0.108	-0.226
C(Location) [T.30] 0.110	-0.0240	0.069	-0.350	0.726	-0.158
C(Location) [T.32] 0.220	0.1022	0.060	1.702	0.089	-0.016
C(Location) [T.33] 0.308	0.1806	0.065	2.787	0.005	0.054
C(Location) [T.34] -0.126	-0.2429	0.059	-4.087	0.000	-0.359
C(Location) [T.35] -0.043	-0.1752	0.067	-2.605	0.009	-0.307
C(Location) [T.36] -0.067	-0.2039	0.070	-2.927	0.003	-0.340
C(Location) [T.38] -0.171	-0.2886	0.060	-4.818	0.000	-0.406
C(Location) [T.39] -0.011	-0.1318	0.062	-2.135	0.033	-0.253
C(Location) [T.40] -0.223	-0.3602	0.070	-5.150	0.000	-0.497
C(Location) [T.41] 0.147	0.0161	0.067	0.240	0.811	-0.115
C(Location) [T.42] 0.122	-0.0881	0.107	-0.820	0.412	-0.298
C(Location) [T.43] 0.222	0.0915	0.067	1.373	0.170	-0.039
C(Location) [T.44] -0.354	-0.4687	0.058	-8.018	0.000	-0.583
C(Location) [T.45] -0.271	-0.3892	0.060	-6.480	0.000	-0.507

C(Location) [T.46]	-0.0503	0.067	-0.750	0.453	-0.182
0.081					
C(Location) [T.47]	-0.0746	0.060	-1.234	0.217	-0.193
0.044					
C(Location) [T.48]	-0.8283	0.061	-13.490	0.000	-0.949
-0.708					
C(Location) [T.49]	-0.4593	0.088	-5.235	0.000	-0.631
-0.287					
Min_Temp	0.1073	0.008	13.493	0.000	0.092
0.123					
Max_Temp	-0.1054	0.008	-13.352	0.000	-0.121
-0.090					
Parameter1_Speed	0.0571	0.003	21.382	0.000	0.052
0.062					
Parameter3_9am	-0.0098	0.004	-2.572	0.010	-0.017
-0.002					
Parameter3_3pm	-0.0582	0.004	-16.280	0.000	-0.065
-0.051					
Parameter4_9am	0.0162	0.002	9.600	0.000	0.013
0.019					
Parameter4_3pm	0.0164	0.002	8.205	0.000	0.012
0.020					
Parameter5_3pm	-0.0208	0.003	-7.438	0.000	-0.026
-0.015					

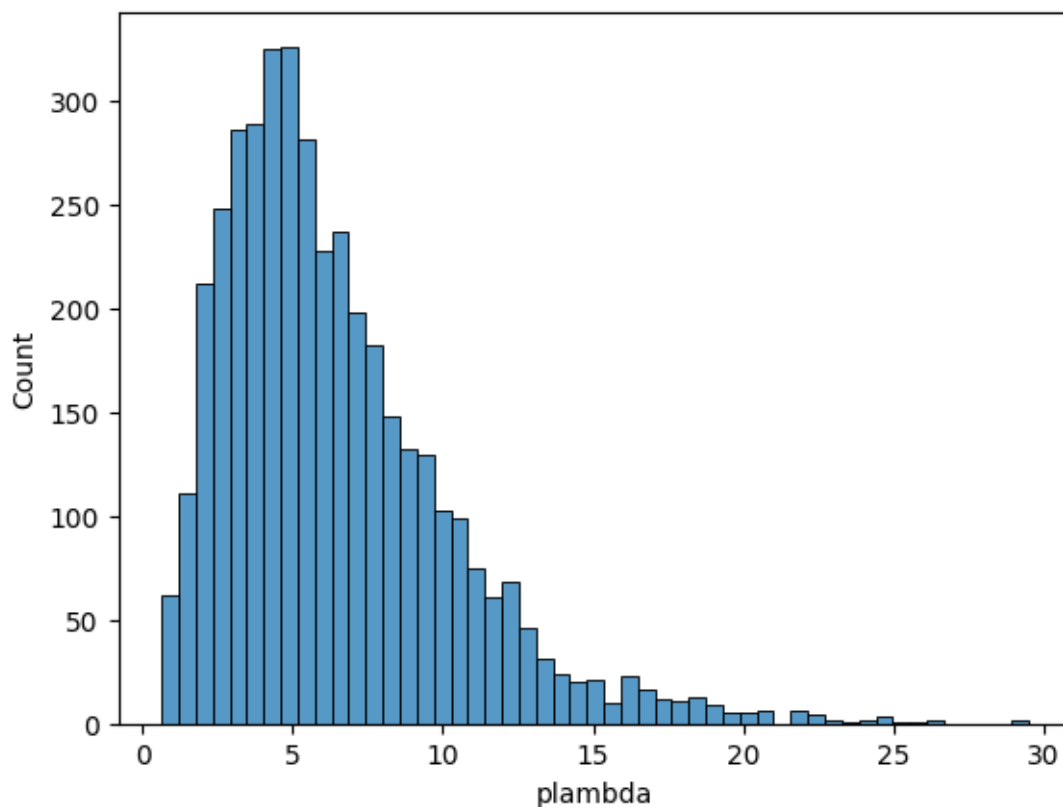
=====

=====

R: Para responder esta pregunta agrupamos la data a nivel mensual y por “Location”. Se estimo el promedio mensual de los datos para cada “Location”, se eliminaron las variables categoricas y se generó un conteo de los fallos por mes en cada “Location” para luego poder aplicarla en el modelo Poisson. En cuanto a los resultados del modelo se obtienen comportamientos similares a los modelos ajustados anteriormente, las variables de temperaturas se comportan de la misma forma, al igual que la de velocidad de viento. En el caso de las “Location” se observa que la mayoría entregan coeficientes negativos, como en el caso de “Location_48” que tiene un coeficiente negativo fuerte (=-0.8283). Se infiere que la ubicación geografica es significativa en cuantos fallos se promedian en el mes.

```
[22]: promedios['plambda'] = poisson.predict()
      sns.histplot(data=promedios, x="plambda")
```

```
[22]: <Axes: xlabel='plambda', ylabel='Count'>
```



7. Determine sobre dispersion en la data y posible valor optimo de alpha para un modelo Bino-
mial Negativa.

```
[23]: #Utilizamos una formula para calcular la sobredispersión y usamos un mu_
      ↪predecido
mu = poisson.predict()
y = promedios["Failure_today"]
aux = ((y - mu)**2 - mu) / mu
auxr = sm.OLS(aux, mu).fit()

print(auxr.summary())
```

OLS Regression Results

```
=====
=====
Dep. Variable:          Failure_today    R-squared (uncentered):
0.001
Model:                  OLS             Adj. R-squared (uncentered):
0.001
Method:                 Least Squares    F-statistic:
5.239
Date:                   Fri, 25 Apr 2025  Prob (F-statistic):
```

```

0.0221
Time:                  12:58:15   Log-Likelihood:
-7162.4
No. Observations:      4076   AIC:
1.433e+04
Df Residuals:          4075   BIC:
1.433e+04
Df Model:              1
Covariance Type:      nonrobust
=====
              coef      std err          t      P>|t|      [0.025      0.975]
-----
x1            -0.0067      0.003      -2.289      0.022      -0.013      -0.001
=====
Omnibus:              3528.464   Durbin-Watson:              1.822
Prob(Omnibus):         0.000   Jarque-Bera (JB):              153178.109
Skew:                  3.948   Prob(JB):              0.00
Kurtosis:              31.975   Cond. No.              1.00
=====

```

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

R: A pesar de obtener un alpha negativo, obtuvimos que es distinto de 0, por lo que concluimos que el modelo Poisson no es adecuado y que existe sobredispersión.

8. Usando la informacion anterior, ejecute un modelo Binomial Negativa para responder a la pregunta 6. Seleccione las variables dependientes a incluir en el modelo final e interprete su significado

```

[24]: binomial_negativa = smf.glm(
      formula="Failure_today ~ C(Location) + Min_Temp + Max_Temp +
      ↪Parameter1_Speed + Parameter3_9am + Parameter3_3pm + Parameter4_9am +
      ↪Parameter4_3pm + Parameter5_3pm",
      data=promedios,
      family=sm.families.NegativeBinomial()
    ).fit()
print(binomial_negativa.summary())

```

Generalized Linear Model Regression Results

```

=====
Dep. Variable:      Failure_today   No. Observations:      4076
Model:              GLM             Df Residuals:          4024
Model Family:      NegativeBinomial  Df Model:              51
Link Function:      Log             Scale:              1.0000
Method:             IRLS            Log-Likelihood:      -11277.

```


Date: Fri, 25 Apr 2025 Deviance: 1069.9
Time: 12:58:15 Pearson chi2: 748.
No. Iterations: 9 Pseudo R-squ. (CS): 0.2767
Covariance Type: nonrobust

```
=====
```

	coef	std err	z	P> z	[0.025
0.975]					

Intercept	22.7994	8.317	2.741	0.006	6.498
39.101					
C(Location) [T.3]	0.1330	0.179	0.744	0.457	-0.218
0.484					
C(Location) [T.4]	0.0414	0.182	0.227	0.820	-0.316
0.399					
C(Location) [T.5]	-0.1932	0.183	-1.054	0.292	-0.553
0.166					
C(Location) [T.6]	-0.3812	0.212	-1.801	0.072	-0.796
0.034					
C(Location) [T.7]	-0.0561	0.181	-0.310	0.757	-0.411
0.299					
C(Location) [T.8]	-0.1003	0.166	-0.604	0.546	-0.425
0.225					
C(Location) [T.9]	-0.2221	0.185	-1.202	0.229	-0.584
0.140					
C(Location) [T.10]	0.0398	0.196	0.203	0.839	-0.344
0.423					
C(Location) [T.11]	0.0274	0.173	0.158	0.874	-0.311
0.366					
C(Location) [T.12]	-0.0735	0.181	-0.406	0.685	-0.429
0.281					
C(Location) [T.13]	-0.2849	0.193	-1.474	0.140	-0.664
0.094					
C(Location) [T.14]	-0.7159	0.183	-3.920	0.000	-1.074
-0.358					
C(Location) [T.15]	-0.2505	0.194	-1.291	0.197	-0.631
0.130					
C(Location) [T.16]	-0.6486	0.167	-3.888	0.000	-0.976
-0.322					
C(Location) [T.17]	-0.9800	0.285	-3.442	0.001	-1.538
-0.422					
C(Location) [T.18]	-0.3504	0.198	-1.772	0.076	-0.738
0.037					
C(Location) [T.19]	-0.3749	0.181	-2.076	0.038	-0.729
-0.021					
C(Location) [T.20]	-0.2376	0.188	-1.264	0.206	-0.606
0.131					

C(Location) [T.21] 0.358	-0.0061	0.186	-0.033	0.974	-0.370
C(Location) [T.22] 0.378	-0.0033	0.195	-0.017	0.986	-0.385
C(Location) [T.23] 0.399	0.0245	0.191	0.128	0.898	-0.350
C(Location) [T.26] 0.234	-0.2113	0.227	-0.929	0.353	-0.657
C(Location) [T.27] -0.436	-0.7723	0.171	-4.506	0.000	-1.108
C(Location) [T.28] -0.419	-0.7806	0.185	-4.230	0.000	-1.142
C(Location) [T.29] 0.297	-0.0426	0.173	-0.246	0.806	-0.383
C(Location) [T.30] 0.273	-0.0846	0.182	-0.464	0.643	-0.442
C(Location) [T.32] 0.308	-0.0147	0.165	-0.089	0.929	-0.338
C(Location) [T.33] 0.417	0.0672	0.179	0.376	0.707	-0.283
C(Location) [T.34] -0.012	-0.3618	0.178	-2.028	0.043	-0.712
C(Location) [T.35] 0.176	-0.1803	0.182	-0.993	0.321	-0.536
C(Location) [T.36] 0.138	-0.2398	0.193	-1.243	0.214	-0.618
C(Location) [T.38] -0.010	-0.3479	0.172	-2.021	0.043	-0.685
C(Location) [T.39] 0.160	-0.1861	0.177	-1.054	0.292	-0.532
C(Location) [T.40] -0.271	-0.6407	0.188	-3.400	0.001	-1.010
C(Location) [T.41] 0.421	0.0678	0.180	0.376	0.707	-0.286
C(Location) [T.42] 0.299	-0.1455	0.227	-0.642	0.521	-0.590
C(Location) [T.43] 0.528	0.1766	0.180	0.984	0.325	-0.175
C(Location) [T.44] -0.265	-0.6051	0.174	-3.486	0.000	-0.945
C(Location) [T.45] -0.046	-0.3826	0.172	-2.229	0.026	-0.719
C(Location) [T.46] 0.298	-0.0652	0.185	-0.352	0.725	-0.429
C(Location) [T.47] 0.151	-0.1997	0.179	-1.116	0.264	-0.550
C(Location) [T.48] -0.660	-0.9981	0.172	-5.791	0.000	-1.336

C(Location) [T.49]	-0.4051	0.193	-2.100	0.036	-0.783
-0.027					
Min_Temp	0.1130	0.021	5.393	0.000	0.072
0.154					
Max_Temp	-0.1058	0.021	-4.996	0.000	-0.147
-0.064					
Parameter1_Speed	0.0646	0.008	8.433	0.000	0.050
0.080					
Parameter3_9am	-0.0075	0.010	-0.734	0.463	-0.028
0.013					
Parameter3_3pm	-0.0657	0.010	-6.671	0.000	-0.085
-0.046					
Parameter4_9am	0.0158	0.004	3.532	0.000	0.007
0.025					
Parameter4_3pm	0.0235	0.006	4.259	0.000	0.013
0.034					
Parameter5_3pm	-0.0230	0.008	-2.875	0.004	-0.039
-0.007					

```
=====
=====
```

```
c:\Users\Nacho\AppData\Local\Programs\Python\Python313\Lib\site-
packages\statsmodels\genmod\families\family.py:1367: ValueWarning: Negative
binomial dispersion parameter alpha not set. Using default value alpha=1.0.
warnings.warn("Negative binomial dispersion parameter alpha not ")
```

R: De acuerdo a los resultados obtenidos en el modelo binomial negativo obtuvimos un menor ajuste de R cuadrado en comparación al modelo Poisson, lo que no es lo esperado dado que al determinar que existía una sobredispersión se esperaba que el modelo binomial negativo se ajustara mejor a los datos. Sin embargo por el lado de los coeficientes se comportan de la misma forma que los coeficientes obtenidos en poisson.

9. Comente los resultados obtenidos en 6, 7 y 8. ¿Cuáles y por qué existen las diferencias entre los resultados?. En su opinión, ¿Cuál sería el más adecuado para responder la pregunta de investigación y por qué? ¿Qué variables resultaron ser robustas a la especificación?

R: En los resultados se obtuvo para el modelo Poisson un Pseudo R cuadrado = 0.3207, en el modelo Binomial Negativo un Pseudo R-squ = 0.2767 y en la pregunta 7 se determinó un alfa negativo (distinto de 0) y significativo ($P > |t| = 0.022$). De acuerdo a los resultados obtenidos es más adecuado el modelo Poisson para responder la pregunta de investigación, dado que reduce en mayor cantidad la incertidumbre a un modelo sin predictores (32.07% > 27.67%). Sin embargo esto no corresponde a lo esperado, como mencioné en la respuesta de la pregunta 8, se esperaba que al confirmar sobredispersión, el modelo de binomial negativa redujera en mayor cantidad la incertidumbre en comparación a Poisson. Por otro lado variables robustas a la especificación a lo largo de los modelos fueron “Min_Temp”, “Max_Temp”, “Parameter1_Speed”, algunas “Location” como “Location_48” que mantuvo constantemente una relación negativa con respecto a los fallos, también las variables estacionales mantuvieron sus comportamientos.