Tarea1_Alvarez_Roldan

May 5, 2025

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
import sklearn
import scipy
from scipy.stats import nbinom
import seaborn as sns
from statsmodels.iolib.summary2 import summary_col
from sklearn.preprocessing import StandardScaler
import datetime

import warnings
warnings.filterwarnings("ignore")

%matplotlib inline
```

Preguntas (todas tienen el mismo puntaje):

1. Cargar la base de datos en el ambiente. Identifique los tipos de datos que se encuentran en la base, realice estadisticas descriptivas sobre las variables importantes (Hint: Revisar la distribuciones, datos faltantes, outliers, etc.) y limpie las variables cuando sea necesario.
2. Ejecute un modelo de probabilidad lineal (MCO) que permita explicar la probabilidad de que un dia se reporte fallo medido por sensor, a partir de las informacion disponible. Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.
3. Ejecute un modelo *probit* para responder a la pregunta 2. Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.
4. Ejecute un modelo *logit* para responder a la pregunta 2. Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.
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 investgación y por qué? ¿Qué variables resultaron ser robustas a la especificación?
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.

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

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.

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 investgación y por qué? ¿Qué variables resultaron ser robustas a la especificación?

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

R: Se leyeron los datos con pandas, para luego realizar una limpieza que consistio en: -Eliminar los NaN de filas que podian resultar muy relevantes para el análisis -Eliminar NaN del resto de filas (Con un porcentaje de NaN igual o menor al 10%) -Crear variables dicotómicas que indicaban si es que faltaban datos en ciertas columnas o no (Las que tenian un porcentaje de NaN mayor al 10%) -Se elimino el "Parameter6_9am" y el "Parameter6_3pm" debido a que ensuciaban mucho la data, y se conservo su dicotomica indicadora -Se simplificaron las variables cardinales de 16 puntos a 4 solamente (N,W,S,E) -Se crearon columnas usando de base la columna "Date" creando así 2 columnas, una llamada "Season" indicando la estación del año y otra llamada "Year" indicando el año en el que se registro el dato -Se transformo la columna de "Failure_today" a dicotomica -Se aplico log(1+x) a la columna "Leakage" para dejar su distribución de forma más uniforme -Se estandarizaron las columnas "Parameter4_9am", "Parameter4_3pm" debido a su alta varianza -Se eliminaron las variables "Location" con los valores 17, 18, 26, 42 para dejar los datos distribuidos de la variable de una forma mucho más uniforme

[2]:
```
df = pd.read_csv('../../data/machine_failure_data.csv', delimiter=",",␣
↪decimal='.')
```

[3]:
```
df
```

[3]:

	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]
```

[4]:
```
#Se eliminaron las filas que tenian un NaN que dificultaban el analisis de los datos

#Se elimino los que no tenian temperatura minima, ya que la temperatura resulta un dato muy relevante a la hora de buscar motivo de falla,
#por lo cual las filas sin la debida información no nos sirven
df = df[df["Min_Temp"].notna()]

#Lo mismo de arriba aplica para la temperatura máxima
df = df[df["Max_Temp"].notna()]

#Las filtraciones son un dato extremadamente importante para detectar fallas, por lo cual deberíamos eliminar las filas con NaN
df = df[df["Leakage"].notna()]
```

[5]:
```
#Se eliminaron las filas que tenian un NaN que dificultaban el analisis de los datos

#Se elimino los que no tenian temperatura minima, ya que la temperatura resulta un dato muy relevante a la hora de buscar motivo de falla,
#por lo cual las filas sin la debida información no nos sirven
df = df[df["Min_Temp"].notna()]

#Lo mismo de arriba aplica para la temperatura máxima
df = df[df["Max_Temp"].notna()]

#Las filtraciones son un dato extremadamente importante para detectar fallas, por lo cual deberíamos eliminar las filas con NaN
df = df[df["Leakage"].notna()]

#Eliminamos los NaN de los parametros medidos, ya que no son tantos datos NaN (Menores o iguales al 10%)
df = df[df["Parameter1_Dir"].notna()]
df = df[df["Parameter1_Speed"].notna()]
df = df[df["Parameter2_9am"].notna()]
df = df[df["Parameter2_3pm"].notna()]
df = df[df["Parameter3_9am"].notna()]
df = df[df["Parameter3_3pm"].notna()]
df = df[df["Parameter4_9am"].notna()]
df = df[df["Parameter4_3pm"].notna()]
```

```
df = df[df["Parameter5_9am"].notna()]
df = df[df["Parameter5_3pm"].notna()]
df = df[df["Parameter7_9am"].notna()]
df = df[df["Parameter7_3pm"].notna()]
```

[6]:
```
#Creamos 2 columnas dicotomicas que nos indican si es que un dato cualquiera␣
 ↪esta presente en las columnas "Parameter6_9am"(38% de NaN)
#y "Parameter6_3pm"(40% de NaN) 1 si es que esta el dato, 0 si es NaN:
nueva_columna=[]
for i in df["Parameter6_9am"]:
    if pd.isna(i):
        nueva_columna.append(0)
    else:
        nueva_columna.append(1)

df["Parameter6_9am_Bin"]=nueva_columna

nueva_columna=[]
for i in df["Electricity"]:
    if pd.isna(i):
        nueva_columna.append(0)
    else:
        nueva_columna.append(1)

df["Parameter6_3pm_Bin"]=nueva_columna

#Eliminamos los parametros 6 ya que ensucian mucho los datos
df = df.drop("Parameter6_3pm", axis=1)
df = df.drop("Parameter6_9am", axis=1)
```

[7]:
```
#Creamos 2 columnas dicotomicas que nos indican si es que un dato cualquiera␣
 ↪esta presente en las columnas "Evaporation" y "Electricity"
#1 si es que esta el dato, 0 si es NaN:
nueva_columna=[]
for i in df["Evaporation"]:
    if pd.isna(i):
        nueva_columna.append(0)
    else:
        nueva_columna.append(1)

df["EvaporationBin"]=nueva_columna

nueva_columna=[]
for i in df["Electricity"]:
    if pd.isna(i):
        nueva_columna.append(0)
    else:
```

```
        nueva_columna.append(1)

df["ElectricityBin"]=nueva_columna
```

[8]:
```
#Se transforman los puntos cardinales con 3 letras y 2 letras a una␣
 ↪solamente(Si es NNW [north-northwest], lo dejamos como N [north])
def transforman_cardinales(columna):
    nueva_columna = []
    for i in columna:
        if isinstance(i, str) and len(i) == 3:
            nueva_columna.append(i[:-2])
        elif isinstance(i, str) and len(i) == 2:
            nueva_columna.append(i[:-1])
        else:
            nueva_columna.append(i)
    return nueva_columna
df["Parameter1_Dir"]=transforman_cardinales(df["Parameter1_Dir"])

df["Parameter2_9am"]=transforman_cardinales(df["Parameter2_9am"])

df["Parameter2_3pm"]=transforman_cardinales(df["Parameter2_3pm"])
```

[9]:
```
#Creamos una columna con los años:
df["Year"] = df["Date"].apply(lambda x: x.split('/')[-1])

#Eliminamos de la data los años que no tengan los 12 meses correspondientes:
df = df[df["Year"] != 2008]
df = df[df["Year"] != 2017]

#Clasificamos por estación
nueva_columna=[]
#Se considera que una estación dura exactamente 3 meses, y que el invierno␣
 ↪empieza en enero
for i in df["Date"]:
    mes = int(i.split('/')[0])
    if mes>=1 and mes<4:
        nueva_columna.append("Winter")
    elif mes>=4 and mes<7:
        nueva_columna.append("Spring")
    elif mes>=7 and mes<10:
        nueva_columna.append("Summer")
    else:
        nueva_columna.append("Autumn")

df["Season"]=nueva_columna
df['Date'] = pd.to_datetime(df['Date'])
#Pasamos year de object a entero para evitar errores
```

```
df["Year"] = df["Year"].astype(int)
```

[10]:
```
nueva_columna=[]
for i in df["Failure_today"]:
    if i=="No":
        nueva_columna.append(0)
    else:
        nueva_columna.append(1)

df["Failure_today"]=nueva_columna
df["Failure_today"].value_counts()
```

[10]:
```
Failure_today
0    87556
1    25369
Name: count, dtype: int64
```

[11]:
```
df["Season"] = df["Season"].astype('category')
df=pd.get_dummies(df, columns=["Season"], drop_first=True)
```

[12]:
```
df["Leakage"] = np.log1p(df["Leakage"])
```

[13]:
```
df_varianza=df.
  ↪drop(["Date","Parameter2_3pm","Parameter2_9am","Parameter1_Dir"], axis=1)
df_varianza.var()
```

[13]:
```
Location              208.907260
Min_Temp               39.114199
Max_Temp               48.758134
Leakage                 0.823964
Evaporation            17.710590
Electricity            14.260742
Parameter1_Speed      177.469652
Parameter3_9am         69.627406
Parameter3_3pm         73.618072
Parameter4_9am        357.648986
Parameter4_3pm        431.332665
Parameter5_9am         49.458223
Parameter5_3pm         48.411286
Parameter7_9am         40.386593
Parameter7_3pm         46.749766
Failure_today           0.174186
Parameter6_9am_Bin      0.221237
Parameter6_3pm_Bin      0.241870
EvaporationBin          0.231601
ElectricityBin          0.241870
Year                    6.401925
```

```
Season_Spring           0.186416
Season_Summer           0.183233
Season_Winter           0.192972
dtype: float64
```

[14]:
```
#Estandarizamos las columnas de alta varianza:
columnas_a_estandarizar=["Parameter4_9am","Parameter4_3pm"]
scaler=StandardScaler()
df[columnas_a_estandarizar] = scaler.fit_transform(df[columnas_a_estandarizar])
```

[15]:
```
#Vamos a eliminar las variables con una frecuencia menor a 2000 para dejar los␣
↪datos de una forma mucho más uniforme:
locaciones_a_eliminar = [17, 18, 26, 42]
indices_a_eliminar = df[df['Location'].isin(locaciones_a_eliminar)].index
df = df.drop(indices_a_eliminar)
```

[16]:
```
df=pd.get_dummies(df, columns=["Parameter1_Dir"], drop_first=True)

df=pd.get_dummies(df, columns=["Parameter2_9am"], drop_first=True)

df=pd.get_dummies(df, columns=["Parameter2_3pm"], drop_first=True)
```

[17]:
```
#Para la matriz de correlaciones del anexo
dfcorrelaciones=df
```

[18]:
```
df=pd.get_dummies(df, columns=["Location"], drop_first=True)
```

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

R: Guíandome por la matriz de correlaciones, elimine del modelo las variables con alta correlación entre si (mayor a 0.8), y elimine la variable leakage (Debido a su predicción practicamente perfecta, lo mismo se realizo para el resto de modelos en el trabajo), de los parametros que tenian p>0.05. Las conclusiones son: -El modelo MCO es capaz de explicar un 28.8% de los datos -Si aumenta en 1 grado la temperatura mínima, la probabilidad de fallo aumenta en un 1.63% -Si disminuye en 1 grado la temperatura máxima, la probabilidad de fallo disminuye en un 1.68% -Si la velocidad del viento aumenta en una milla por hora,a probabilidad de fallo aumenta en un 0.053% -Y asi se realizo la misma interpretación con el resto de parametros significativos, multiplicando su valor por 100 e interpretando el resultado como su variación porcentual. -Cabe destacar que cada locación tenia influencias completamente distintas en la explicación de la falla, por lo cual habria que revisar la locación de interes en cada paso en particular para realizar las estimaciones.

[19]:
```
#Regresion excluyendo variables de alta correlacion y las con NaN
y=df['Failure_today']
X=df.drop(['Failure_today','Date','Leakage',␣
↪'Parameter5_3pm','Parameter7_3pm','Parameter7_9am','Parameter6_3pm_Bin','Evaporation','Elec
↪axis=1)
```

```
X = X.astype(float)
X=sm.add_constant(X)
model = sm.OLS(y, X)
results = model.fit(cov_type='HC0')
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          Failure_today   R-squared:                       0.288
Model:                            OLS   Adj. R-squared:                  0.288
Method:                 Least Squares   F-statistic:                     714.2
Date:                Thu, 24 Apr 2025   Prob (F-statistic):               0.00
Time:                        21:55:24   Log-Likelihood:                -40899.
No. Observations:              107753   AIC:                          8.193e+04
Df Residuals:                  107689   BIC:                          8.254e+04
Df Model:                          63
Covariance Type:                  HC0
==============================================================================
======
                         coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
------
const                  7.8930      0.971      8.130      0.000       5.990
9.796
Min_Temp               0.0163      0.000     39.287      0.000       0.015
0.017
Max_Temp              -0.0168      0.000    -40.424      0.000      -0.018
-0.016
Parameter1_Speed       0.0051      0.000     35.075      0.000       0.005
0.005
Parameter3_9am         0.0033      0.000     17.833      0.000       0.003
0.004
Parameter3_3pm        -0.0040      0.000    -20.847      0.000      -0.004
-0.004
Parameter4_9am         0.1520      0.002     84.693      0.000       0.148
0.155
Parameter4_3pm         0.0133      0.002      5.779      0.000       0.009
0.018
Parameter5_9am        -0.0087      0.000    -39.075      0.000      -0.009
-0.008
Parameter6_9am_Bin     0.0386      0.004      9.573      0.000       0.031
0.046
EvaporationBin        -0.0165      0.005     -3.178      0.001      -0.027
-0.006
ElectricityBin        -0.0019      0.005     -0.375      0.707      -0.012
0.008
Year                   0.0006      0.000      1.329      0.184      -0.000
```

```
0.002
Season_Spring          -0.0396      0.004    -11.249      0.000      -0.047
-0.033
Season_Summer          -0.0104      0.004     -2.733      0.006      -0.018
-0.003
Season_Winter          -0.0476      0.003    -14.533      0.000      -0.054
-0.041
Parameter1_Dir_N       -0.0181      0.004     -4.813      0.000      -0.025
-0.011
Parameter1_Dir_S        0.0047      0.004      1.274      0.203      -0.003
0.012
Parameter1_Dir_W        0.0050      0.004      1.142      0.253      -0.004
0.014
Parameter2_9am_N        0.0017      0.003      0.524      0.601      -0.005
0.008
Parameter2_9am_S        0.0326      0.003      9.801      0.000       0.026
0.039
Parameter2_9am_W        0.0542      0.004     12.568      0.000       0.046
0.063
Parameter2_3pm_N       -0.0083      0.004     -2.259      0.024      -0.016
-0.001
Parameter2_3pm_S        0.0301      0.004      8.313      0.000       0.023
0.037
Parameter2_3pm_W        0.0382      0.004      8.827      0.000       0.030
0.047
Location_3             -0.0882      0.010     -8.658      0.000      -0.108
-0.068
Location_4              0.0638      0.009      7.020      0.000       0.046
0.082
Location_5             -0.1062      0.010    -10.230      0.000      -0.127
-0.086
Location_6             -0.2431      0.011    -21.602      0.000      -0.265
-0.221
Location_7             -0.1272      0.010    -12.645      0.000      -0.147
-0.108
Location_8             -0.0455      0.010     -4.463      0.000      -0.065
-0.026
Location_9             -0.1061      0.011     -9.614      0.000      -0.128
-0.084
Location_10            -0.1086      0.010    -10.862      0.000      -0.128
-0.089
Location_11            -0.0614      0.010     -6.297      0.000      -0.081
-0.042
Location_12            -0.0568      0.011     -5.302      0.000      -0.078
-0.036
Location_13            -0.1183      0.011    -10.866      0.000      -0.140
-0.097
Location_14            -0.1387      0.011    -12.912      0.000      -0.160
```

-0.118
Location_15 -0.0941 0.010 -9.142 0.000 -0.114
-0.074
Location_16 -0.1577 0.010 -15.585 0.000 -0.178
-0.138
Location_19 -0.1382 0.011 -12.401 0.000 -0.160
-0.116
Location_20 -0.1845 0.010 -17.623 0.000 -0.205
-0.164
Location_21 -0.1241 0.009 -13.137 0.000 -0.143
-0.106
Location_22 -0.0834 0.009 -8.881 0.000 -0.102
-0.065
Location_23 -0.1113 0.011 -10.483 0.000 -0.132
-0.090
Location_27 -0.1652 0.011 -15.712 0.000 -0.186
-0.145
Location_28 -0.1843 0.011 -16.980 0.000 -0.206
-0.163
Location_29 -0.1054 0.010 -10.660 0.000 -0.125
-0.086
Location_30 -0.0585 0.011 -5.489 0.000 -0.079
-0.038
Location_32 -0.0596 0.010 -6.200 0.000 -0.078
-0.041
Location_33 -0.0596 0.010 -6.117 0.000 -0.079
-0.040
Location_34 -0.1299 0.011 -11.979 0.000 -0.151
-0.109
Location_35 -0.0989 0.011 -9.238 0.000 -0.120
-0.078
Location_36 -0.2211 0.011 -20.738 0.000 -0.242
-0.200
Location_38 -0.1319 0.011 -11.771 0.000 -0.154
-0.110
Location_39 -0.1193 0.011 -11.336 0.000 -0.140
-0.099
Location_40 -0.1486 0.010 -14.583 0.000 -0.169
-0.129
Location_41 -0.0726 0.010 -7.047 0.000 -0.093
-0.052
Location_43 -0.0849 0.010 -8.643 0.000 -0.104
-0.066
Location_44 -0.1115 0.011 -10.242 0.000 -0.133
-0.090
Location_45 -0.1668 0.011 -15.856 0.000 -0.187
-0.146
Location_46 -0.0963 0.011 -8.581 0.000 -0.118

```
-0.074
Location_47           -0.0578      0.011     -5.367      0.000      -0.079
-0.037
Location_48           -0.2004      0.010    -19.176      0.000      -0.221
-0.180
Location_49           -0.1239      0.009    -13.581      0.000      -0.142
-0.106
==============================================================================
Omnibus:                     8344.566   Durbin-Watson:                   1.800
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            10232.482
Skew:                           0.746   Prob(JB):                         0.00
Kurtosis:                       2.765   Cond. No.                     2.03e+06
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
[2] The condition number is large, 2.03e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

3. Ejecute un modelo *probit* para responder a la pregunta 2. Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.

R: Para el modelo probit se utilizaron las mismas variables que en el MCO, para poder comparar la efectividad de los modelos y sus diferencias en 5.

Las conclusiones son: -El modelo probit es capaz de explicar un 32.74% de los datos -Si aumenta en 1 grado la temperatura mínima, la probabilidad de fallo aumenta en un 1.92% -Si disminuye en 1 grado la temperatura máxima, la probabilidad de fallo disminuye en un 2.40% -De el resto de parametros cabe destacar el Parameter4_9am, que al subir en 1 unidad tiene un efecto del aumentar la probabilidad de fallo en un 16.04%, por lo que es una variable relevante. -La dirección del viento tambien resulta relevante, teniendo direccion “West” a las 9am provoca un aumento 4.92% de falla, y teniendo la direccion “South” a la misma hora provocaba un aumento en la probabilidad de falla de un 3.95% -Se detectan porcentajes de falla relevantes respecto a las estaciones, con un 5.69% de probablidad disminución de falla en “Spring” y 3.93% en “Summer” y 3.62% en “Winter”, por lo que es probable que las fallas sean más propensas a suceder en “Autumn”. -Y asi se realizo la misma interpretación con el resto de parametros significativos, multiplicando su valor por 100 e interpretando el resultado como su variación porcentual. -Al igual que en el modelo anterior cada locación tiene influencias completamente distintas en la explicación de la falla, por lo cual habria que revisar la locación de interes en cada paso en particular para realizar las estimaciones.

[20]:
```
model = sm.Probit(y, X)
probit_model = model.fit(cov_type='HC0')
print(probit_model.summary())

mfxp = probit_model.get_margeff()
print(mfxp.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.360727
         Iterations 7
```

Probit Regression Results

Dep. Variable:	Failure_today	No. Observations:	107753
Model:	Probit	Df Residuals:	107689
Method:	MLE	Df Model:	63
Date:	Thu, 24 Apr 2025	Pseudo R-squ.:	0.3274
Time:	21:55:25	Log-Likelihood:	-38869.
converged:	True	LL-Null:	-57786.
Covariance Type:	HC0	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	28.3195	4.594	6.164	0.000	19.315	37.324
Min_Temp	0.0948	0.002	43.140	0.000	0.091	0.099
Max_Temp	-0.1185	0.002	-49.344	0.000	-0.123	-0.114
Parameter1_Speed	0.0193	0.001	30.673	0.000	0.018	0.020
Parameter3_9am	0.0114	0.001	12.895	0.000	0.010	0.013
Parameter3_3pm	-0.0133	0.001	-14.860	0.000	-0.015	-0.012
Parameter4_9am	0.7921	0.010	80.535	0.000	0.773	0.811
Parameter4_3pm	-0.0509	0.010	-5.157	0.000	-0.070	-0.032
Parameter5_9am	-0.0325	0.001	-33.477	0.000	-0.034	-0.031
Parameter6_9am_Bin	0.1326	0.022	6.070	0.000	0.090	0.175
EvaporationBin	-0.0163	0.026	-0.639	0.523	-0.066	0.034
ElectricityBin	-0.0395	0.026	-1.514	0.130	-0.091	0.012
Year	0.0024	0.002	1.063	0.288	-0.002	0.007
Season_Spring	-0.2810	0.017	-16.660	0.000	-0.314	-0.248
Season_Summer	-0.1939	0.019	-10.416	0.000	-0.230	-0.157
Season_Winter	-0.1787	0.016	-10.871	0.000	-0.211	-0.146
Parameter1_Dir_N	-0.1290	0.021	-6.165	0.000	-0.170	

```
-0.088
Parameter1_Dir_S      -0.0057      0.019     -0.300      0.764      -0.043
0.031
Parameter1_Dir_W       0.0102      0.022      0.463      0.643      -0.033
0.054
Parameter2_9am_N       0.0204      0.020      1.037      0.300      -0.018
0.059
Parameter2_9am_S       0.1949      0.018     10.686      0.000       0.159
0.231
Parameter2_9am_W       0.2432      0.021     11.677      0.000       0.202
0.284
Parameter2_3pm_N      -0.0556      0.020     -2.763      0.006      -0.095
-0.016
Parameter2_3pm_S       0.0836      0.018      4.562      0.000       0.048
0.119
Parameter2_3pm_W       0.1041      0.022      4.734      0.000       0.061
0.147
Location_3            -0.3061      0.055     -5.590      0.000      -0.413
-0.199
Location_4             0.1199      0.066      1.828      0.068      -0.009
0.249
Location_5            -0.2412      0.050     -4.867      0.000      -0.338
-0.144
Location_6            -1.1311      0.056    -20.283      0.000      -1.240
-1.022
Location_7            -0.5417      0.054    -10.026      0.000      -0.648
-0.436
Location_8             0.1982      0.050      3.972      0.000       0.100
0.296
Location_9            -0.0932      0.050     -1.853      0.064      -0.192
0.005
Location_10           -0.3041      0.052     -5.851      0.000      -0.406
-0.202
Location_11           -0.2937      0.060     -4.879      0.000      -0.412
-0.176
Location_12            0.0089      0.049      0.180      0.857      -0.088
0.106
Location_13           -0.5498      0.047    -11.579      0.000      -0.643
-0.457
Location_14           -0.1657      0.053     -3.154      0.002      -0.269
-0.063
Location_15           -0.0886      0.048     -1.842      0.065      -0.183
0.006
Location_16           -0.4681      0.049     -9.582      0.000      -0.564
-0.372
Location_19           -0.3824      0.052     -7.396      0.000      -0.484
-0.281
Location_20           -0.6634      0.052    -12.874      0.000      -0.764
```

```
-0.562
Location_21          -0.7135      0.057    -12.612      0.000      -0.824
-0.603
Location_22          -0.1219      0.056     -2.182      0.029      -0.231
-0.012
Location_23          -0.4481      0.050     -8.984      0.000      -0.546
-0.350
Location_27          -0.5157      0.047    -10.953      0.000      -0.608
-0.423
Location_28          -0.5822      0.049    -11.983      0.000      -0.677
-0.487
Location_29          -0.6113      0.055    -11.142      0.000      -0.719
-0.504
Location_30          -0.0382      0.058     -0.661      0.509      -0.152
0.075
Location_32          -0.0744      0.050     -1.480      0.139      -0.173
0.024
Location_33          -0.0491      0.052     -0.945      0.345      -0.151
0.053
Location_34          -0.5496      0.048    -11.340      0.000      -0.645
-0.455
Location_35          -0.2480      0.052     -4.763      0.000      -0.350
-0.146
Location_36          -0.7670      0.052    -14.831      0.000      -0.868
-0.666
Location_38          -0.3053      0.050     -6.090      0.000      -0.404
-0.207
Location_39          -0.2870      0.051     -5.591      0.000      -0.388
-0.186
Location_40          -0.2634      0.053     -4.991      0.000      -0.367
-0.160
Location_41          -0.1401      0.050     -2.787      0.005      -0.239
-0.042
Location_43          -0.3064      0.055     -5.547      0.000      -0.415
-0.198
Location_44          -0.3558      0.047     -7.568      0.000      -0.448
-0.264
Location_45          -0.6361      0.051    -12.426      0.000      -0.736
-0.536
Location_46          -0.1585      0.052     -3.031      0.002      -0.261
-0.056
Location_47          -0.0991      0.049     -2.040      0.041      -0.194
-0.004
Location_48          -0.6641      0.050    -13.207      0.000      -0.763
-0.566
Location_49          -0.8510      0.064    -13.301      0.000      -0.976
-0.726
==============================================================================
```

```
======
       Probit Marginal Effects
=====================================
Dep. Variable:          Failure_today
Method:                          dydx
At:                           overall
==============================================================================
======
                        dy/dx    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
------
Min_Temp                0.0192      0.000     44.166      0.000       0.018
0.020
Max_Temp               -0.0240      0.000    -51.730      0.000      -0.025
-0.023
Parameter1_Speed        0.0039      0.000     31.174      0.000       0.004
0.004
Parameter3_9am          0.0023      0.000     12.931      0.000       0.002
0.003
Parameter3_3pm         -0.0027      0.000    -14.914      0.000      -0.003
-0.002
Parameter4_9am          0.1604      0.002     93.030      0.000       0.157
0.164
Parameter4_3pm         -0.0103      0.002     -5.163      0.000      -0.014
-0.006
Parameter5_9am         -0.0066      0.000    -34.048      0.000      -0.007
-0.006
Parameter6_9am_Bin      0.0268      0.004      6.076      0.000       0.018
0.035
EvaporationBin         -0.0033      0.005     -0.639      0.523      -0.013
0.007
ElectricityBin         -0.0080      0.005     -1.514      0.130      -0.018
0.002
Year                    0.0005      0.000      1.063      0.288      -0.000
0.001
Season_Spring          -0.0569      0.003    -16.692      0.000      -0.064
-0.050
Season_Summer          -0.0393      0.004    -10.425      0.000      -0.047
-0.032
Season_Winter          -0.0362      0.003    -10.871      0.000      -0.043
-0.030
Parameter1_Dir_N       -0.0261      0.004     -6.172      0.000      -0.034
-0.018
Parameter1_Dir_S       -0.0011      0.004     -0.300      0.764      -0.009
0.006
Parameter1_Dir_W        0.0021      0.004      0.463      0.643      -0.007
0.011
```

Parameter2_9am_N	0.0041	0.004	1.037	0.300	-0.004 0.012
Parameter2_9am_S	0.0395	0.004	10.696	0.000	0.032 0.047
Parameter2_9am_W	0.0492	0.004	11.696	0.000	0.041 0.057
Parameter2_3pm_N	-0.0113	0.004	-2.763	0.006	-0.019 -0.003
Parameter2_3pm_S	0.0169	0.004	4.563	0.000	0.010 0.024
Parameter2_3pm_W	0.0211	0.004	4.735	0.000	0.012 0.030
Location_3	-0.0620	0.011	-5.599	0.000	-0.084 -0.040
Location_4	0.0243	0.013	1.828	0.068	-0.002 0.050
Location_5	-0.0488	0.010	-4.870	0.000	-0.068 -0.029
Location_6	-0.2290	0.011	-20.525	0.000	-0.251 -0.207
Location_7	-0.1097	0.011	-10.057	0.000	-0.131 -0.088
Location_8	0.0401	0.010	3.973	0.000	0.020 0.060
Location_9	-0.0189	0.010	-1.853	0.064	-0.039 0.001
Location_10	-0.0616	0.011	-5.859	0.000	-0.082 -0.041
Location_11	-0.0595	0.012	-4.885	0.000	-0.083 -0.036
Location_12	0.0018	0.010	0.180	0.857	-0.018 0.021
Location_13	-0.1113	0.010	-11.617	0.000	-0.130 -0.093
Location_14	-0.0335	0.011	-3.153	0.002	-0.054 -0.013
Location_15	-0.0179	0.010	-1.842	0.065	-0.037 0.001
Location_16	-0.0948	0.010	-9.615	0.000	-0.114 -0.075
Location_19	-0.0774	0.010	-7.410	0.000	-0.098 -0.057
Location_20	-0.1343	0.010	-12.932	0.000	-0.155 -0.114
Location_21	-0.1445	0.011	-12.662	0.000	-0.167 -0.122
Location_22	-0.0247	0.011	-2.182	0.029	-0.047 -0.003

```
Location_23           -0.0907      0.010     -9.006      0.000      -0.110
-0.071
Location_27           -0.1044      0.010    -10.978      0.000      -0.123
-0.086
Location_28           -0.1179      0.010    -12.019      0.000      -0.137
-0.099
Location_29           -0.1238      0.011    -11.189      0.000      -0.145
-0.102
Location_30           -0.0077      0.012     -0.661      0.509      -0.031
0.015
Location_32           -0.0151      0.010     -1.480      0.139      -0.035
0.005
Location_33           -0.0099      0.011     -0.945      0.345      -0.031
0.011
Location_34           -0.1113      0.010    -11.379      0.000      -0.130
-0.092
Location_35           -0.0502      0.011     -4.766      0.000      -0.071
-0.030
Location_36           -0.1553      0.010    -14.927      0.000      -0.176
-0.135
Location_38           -0.0618      0.010     -6.096      0.000      -0.082
-0.042
Location_39           -0.0581      0.010     -5.596      0.000      -0.078
-0.038
Location_40           -0.0533      0.011     -4.990      0.000      -0.074
-0.032
Location_41           -0.0284      0.010     -2.788      0.005      -0.048
-0.008
Location_43           -0.0620      0.011     -5.556      0.000      -0.084
-0.040
Location_44           -0.0720      0.010     -7.579      0.000      -0.091
-0.053
Location_45           -0.1288      0.010    -12.482      0.000      -0.149
-0.109
Location_46           -0.0321      0.011     -3.032      0.002      -0.053
-0.011
Location_47           -0.0201      0.010     -2.040      0.041      -0.039
-0.001
Location_48           -0.1345      0.010    -13.253      0.000      -0.154
-0.115
Location_49           -0.1723      0.013    -13.381      0.000      -0.198
-0.147
==============================================================================
======
```

4. Ejecute un modelo *logit* para responder a la pregunta 2. Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.

R: Para el modelo logit se utilizaron las mismas variables que en los 2 modelos anteriores, para

poder compararlos en 5.

Las conclusiones son: -El modelo probit es capaz de explicar un 32.96% de los datos -Si aumenta en 1 grado la temperatura mínima, la probabilidad de fallo aumenta en un 1.94% -Si disminuye en 1 grado la temperatura máxima, la probabilidad de fallo disminuye en un 2.49% -De el resto de parametros cabe destacar que otra vez el Parameter4_9am es altamente influyente, al subir en 1 unidad este tiene un efecto del aumentar la probabilidad de fallo en un 16.31%, por lo que es una variable relevante. -La dirección del viento vuelve a ser relevante mayormente en las mismas direcciones analizadas anteriormente y a la misma hora (9am), teniendo direccion "West" provoca un aumento 4.93% de falla, y la direccion "South" de un provoca un aumento del 3.93% en la probabilidad de falla -Acá igualmente se detectan porcentajes de falla relevantes respecto a las estaciones, con un 5.40% de probablidad disminución de falla en "Spring" y 3.82% en "Summer" y 3.33% en "Winter", por lo que es probable que las fallas sean más propensas a suceder en "Autumn". -Y asi se realizo la misma interpretación con el resto de parametros significativos, multiplicando su valor por 100 e interpretando el resultado como su variación porcentual. -Al igual que los modelos anteriores, cada locación tiene influencias completamente distintas en la explicación de la falla, por lo cual habria que revisar la locación de interes en cada paso en particular caso para realizar las estimaciones.

[21]:
```
model = sm.Logit(y, X)
logit_model = model.fit(cov_type='HC0')
print(logit_model.summary())

mfxl = logit_model.get_margeff()
print(mfxl.summary())

params = logit_model.params
conf = logit_model.conf_int()
conf['Odds Ratio'] = params
conf.columns = ['Odds Ratio', '5%', '95%']
print("Odds Ratios")
print(np.exp(conf).iloc[1:17 , ])
```

```
Optimization terminated successfully.
         Current function value: 0.359504
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:          Failure_today   No. Observations:               107753
Model:                          Logit   Df Residuals:                   107689
Method:                           MLE   Df Model:                           63
Date:                Thu, 24 Apr 2025   Pseudo R-squ.:                  0.3296
Time:                        21:55:47   Log-Likelihood:                -38738.
converged:                       True   LL-Null:                       -57786.
Covariance Type:                  HC0   LLR p-value:                     0.000
==============================================================================
======
                           coef    std err          z      P>|z|      [0.025
0.975]
```

```
------------------------------------------------------------------------------
------
const                50.2301      8.114      6.190      0.000      34.326
66.134
Min_Temp              0.1703      0.004     43.863      0.000       0.163
0.178
Max_Temp             -0.2184      0.004    -51.181      0.000      -0.227
-0.210
Parameter1_Speed      0.0337      0.001     30.397      0.000       0.032
0.036
Parameter3_9am        0.0193      0.002     12.331      0.000       0.016
0.022
Parameter3_3pm       -0.0224      0.002    -14.099      0.000      -0.026
-0.019
Parameter4_9am        1.4329      0.017     82.350      0.000       1.399
1.467
Parameter4_3pm       -0.1093      0.017     -6.321      0.000      -0.143
-0.075
Parameter5_9am       -0.0562      0.002    -32.873      0.000      -0.060
-0.053
Parameter6_9am_Bin    0.2559      0.039      6.490      0.000       0.179
0.333
EvaporationBin       -0.0307      0.045     -0.681      0.496      -0.119
0.058
ElectricityBin       -0.0704      0.046     -1.527      0.127      -0.161
0.020
Year                  0.0036      0.004      0.906      0.365      -0.004
0.011
Season_Spring        -0.4743      0.030    -15.748      0.000      -0.533
-0.415
Season_Summer        -0.3360      0.033    -10.111      0.000      -0.401
-0.271
Season_Winter        -0.2896      0.029     -9.820      0.000      -0.347
-0.232
Parameter1_Dir_N     -0.2435      0.037     -6.558      0.000      -0.316
-0.171
Parameter1_Dir_S     -0.0257      0.033     -0.773      0.440      -0.091
0.040
Parameter1_Dir_W      0.0022      0.039      0.055      0.956      -0.074
0.079
Parameter2_9am_N      0.0344      0.035      0.985      0.324      -0.034
0.103
Parameter2_9am_S      0.3452      0.032     10.640      0.000       0.282
0.409
Parameter2_9am_W      0.4332      0.037     11.737      0.000       0.361
0.506
Parameter2_3pm_N     -0.1042      0.036     -2.911      0.004      -0.174
-0.034
```

Parameter2_3pm_S	0.1349	0.032	4.163	0.000	0.071 0.198
Parameter2_3pm_W	0.1695	0.039	4.366	0.000	0.093 0.246
Location_3	-0.6274	0.098	-6.422	0.000	-0.819 -0.436
Location_4	0.1240	0.117	1.060	0.289	-0.105 0.353
Location_5	-0.4032	0.089	-4.552	0.000	-0.577 -0.230
Location_6	-2.0994	0.098	-21.463	0.000	-2.291 -1.908
Location_7	-1.0222	0.096	-10.636	0.000	-1.211 -0.834
Location_8	0.3991	0.089	4.477	0.000	0.224 0.574
Location_9	-0.0819	0.089	-0.916	0.360	-0.257 0.093
Location_10	-0.5851	0.094	-6.257	0.000	-0.768 -0.402
Location_11	-0.6320	0.108	-5.862	0.000	-0.843 -0.421
Location_12	0.0409	0.088	0.465	0.642	-0.132 0.213
Location_13	-0.9950	0.083	-11.927	0.000	-1.159 -0.831
Location_14	-0.2003	0.094	-2.132	0.033	-0.385 -0.016
Location_15	-0.0889	0.085	-1.041	0.298	-0.256 0.079
Location_16	-0.8904	0.088	-10.144	0.000	-1.062 -0.718
Location_19	-0.7054	0.092	-7.647	0.000	-0.886 -0.525
Location_20	-1.2132	0.092	-13.172	0.000	-1.394 -1.033
Location_21	-1.3238	0.101	-13.127	0.000	-1.521 -1.126
Location_22	-0.2502	0.103	-2.431	0.015	-0.452 -0.048
Location_23	-0.8303	0.089	-9.367	0.000	-1.004 -0.657
Location_27	-0.8991	0.084	-10.721	0.000	-1.063 -0.735
Location_28	-1.0243	0.087	-11.824	0.000	-1.194 -0.855
Location_29	-1.1685	0.098	-11.969	0.000	-1.360 -0.977

```
Location_30           -0.0835      0.103     -0.813      0.416      -0.285
0.118
Location_32           -0.1183      0.089     -1.323      0.186      -0.293
0.057
Location_33           -0.0802      0.093     -0.866      0.386      -0.262
0.101
Location_34           -1.0029      0.086    -11.618      0.000      -1.172
-0.834
Location_35           -0.4201      0.093     -4.524      0.000      -0.602
-0.238
Location_36           -1.4103      0.092    -15.276      0.000      -1.591
-1.229
Location_38           -0.5308      0.089     -5.953      0.000      -0.706
-0.356
Location_39           -0.5272      0.093     -5.668      0.000      -0.710
-0.345
Location_40           -0.3668      0.094     -3.883      0.000      -0.552
-0.182
Location_41           -0.2650      0.090     -2.955      0.003      -0.441
-0.089
Location_43           -0.6425      0.100     -6.456      0.000      -0.838
-0.447
Location_44           -0.6296      0.084     -7.532      0.000      -0.793
-0.466
Location_45           -1.1782      0.091    -12.950      0.000      -1.357
-1.000
Location_46           -0.2912      0.093     -3.115      0.002      -0.474
-0.108
Location_47           -0.1712      0.086     -1.989      0.047      -0.340
-0.002
Location_48           -1.1836      0.091    -13.063      0.000      -1.361
-1.006
Location_49           -1.6147      0.112    -14.363      0.000      -1.835
-1.394
==============================================================================
======
        Logit Marginal Effects
=====================================
Dep. Variable:          Failure_today
Method:                          dydx
At:                           overall
==============================================================================
======
                        dy/dx    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
------
Min_Temp                0.0194      0.000     44.957      0.000       0.019
```

```
0.020
Max_Temp              -0.0249      0.000    -53.392      0.000      -0.026
-0.024
Parameter1_Speed       0.0038      0.000     31.036      0.000       0.004
0.004
Parameter3_9am         0.0022      0.000     12.366      0.000       0.002
0.003
Parameter3_3pm        -0.0026      0.000    -14.156      0.000      -0.003
-0.002
Parameter4_9am         0.1631      0.002     96.208      0.000       0.160
0.166
Parameter4_3pm        -0.0124      0.002     -6.326      0.000      -0.016
-0.009
Parameter5_9am        -0.0064      0.000    -33.454      0.000      -0.007
-0.006
Parameter6_9am_Bin     0.0291      0.004      6.498      0.000       0.020
0.038
EvaporationBin        -0.0035      0.005     -0.681      0.496      -0.014
0.007
ElectricityBin        -0.0080      0.005     -1.527      0.127      -0.018
0.002
Year                   0.0004      0.000      0.906      0.365      -0.000
0.001
Season_Spring         -0.0540      0.003    -15.756      0.000      -0.061
-0.047
Season_Summer         -0.0382      0.004    -10.109      0.000      -0.046
-0.031
Season_Winter         -0.0330      0.003     -9.823      0.000      -0.040
-0.026
Parameter1_Dir_N      -0.0277      0.004     -6.565      0.000      -0.036
-0.019
Parameter1_Dir_S      -0.0029      0.004     -0.773      0.440      -0.010
0.005
Parameter1_Dir_W       0.0002      0.004      0.055      0.956      -0.008
0.009
Parameter2_9am_N       0.0039      0.004      0.985      0.324      -0.004
0.012
Parameter2_9am_S       0.0393      0.004     10.646      0.000       0.032
0.047
Parameter2_9am_W       0.0493      0.004     11.756      0.000       0.041
0.058
Parameter2_3pm_N      -0.0119      0.004     -2.912      0.004      -0.020
-0.004
Parameter2_3pm_S       0.0154      0.004      4.165      0.000       0.008
0.023
Parameter2_3pm_W       0.0193      0.004      4.368      0.000       0.011
0.028
Location_3            -0.0714      0.011     -6.433      0.000      -0.093
```

-0.050
Location_4 0.0141 0.013 1.060 0.289 -0.012
0.040
Location_5 -0.0459 0.010 -4.555 0.000 -0.066
-0.026
Location_6 -0.2389 0.011 -21.726 0.000 -0.260
-0.217
Location_7 -0.1163 0.011 -10.670 0.000 -0.138
-0.095
Location_8 0.0454 0.010 4.478 0.000 0.026
0.065
Location_9 -0.0093 0.010 -0.916 0.360 -0.029
0.011
Location_10 -0.0666 0.011 -6.268 0.000 -0.087
-0.046
Location_11 -0.0719 0.012 -5.870 0.000 -0.096
-0.048
Location_12 0.0047 0.010 0.465 0.642 -0.015
0.024
Location_13 -0.1132 0.009 -11.968 0.000 -0.132
-0.095
Location_14 -0.0228 0.011 -2.132 0.033 -0.044
-0.002
Location_15 -0.0101 0.010 -1.041 0.298 -0.029
0.009
Location_16 -0.1013 0.010 -10.187 0.000 -0.121
-0.082
Location_19 -0.0803 0.010 -7.663 0.000 -0.101
-0.060
Location_20 -0.1381 0.010 -13.238 0.000 -0.159
-0.118
Location_21 -0.1507 0.011 -13.185 0.000 -0.173
-0.128
Location_22 -0.0285 0.012 -2.432 0.015 -0.051
-0.006
Location_23 -0.0945 0.010 -9.391 0.000 -0.114
-0.075
Location_27 -0.1023 0.010 -10.751 0.000 -0.121
-0.084
Location_28 -0.1166 0.010 -11.867 0.000 -0.136
-0.097
Location_29 -0.1330 0.011 -12.015 0.000 -0.155
-0.111
Location_30 -0.0095 0.012 -0.813 0.416 -0.032
0.013
Location_32 -0.0135 0.010 -1.324 0.186 -0.033
0.006
Location_33 -0.0091 0.011 -0.866 0.386 -0.030

```
0.012
Location_34          -0.1141      0.010    -11.658      0.000      -0.133
-0.095
Location_35          -0.0478      0.011     -4.527      0.000      -0.069
-0.027
Location_36          -0.1605      0.010    -15.385      0.000      -0.181
-0.140
Location_38          -0.0604      0.010     -5.960      0.000      -0.080
-0.041
Location_39          -0.0600      0.011     -5.675      0.000      -0.081
-0.039
Location_40          -0.0417      0.011     -3.883      0.000      -0.063
-0.021
Location_41          -0.0302      0.010     -2.956      0.003      -0.050
-0.010
Location_43          -0.0731      0.011     -6.467      0.000      -0.095
-0.051
Location_44          -0.0717      0.009     -7.545      0.000      -0.090
-0.053
Location_45          -0.1341      0.010    -13.015      0.000      -0.154
-0.114
Location_46          -0.0331      0.011     -3.117      0.002      -0.054
-0.012
Location_47          -0.0195      0.010     -1.989      0.047      -0.039
-0.000
Location_48          -0.1347      0.010    -13.118      0.000      -0.155
-0.115
Location_49          -0.1838      0.013    -14.434      0.000      -0.209
-0.159
==============================================================================
======
Odds Ratios
                    Odds Ratio        5%       95%
Min_Temp              1.176646  1.194689  1.185633
Max_Temp              0.797135  0.810579  0.803828
Parameter1_Speed      1.032048  1.036546  1.034294
Parameter3_9am        1.016400  1.022669  1.019530
Parameter3_3pm        0.974789  0.980883  0.977831
Parameter4_9am        4.050287  4.336182  4.190797
Parameter4_3pm        0.866538  0.927336  0.896422
Parameter5_9am        0.942207  0.948540  0.945368
Parameter6_9am_Bin    1.195575  1.395407  1.291632
EvaporationBin        0.887689  1.059357  0.969732
ElectricityBin        0.851510  1.020161  0.932029
Year                  0.995812  1.011471  1.003611
Season_Spring         0.586659  0.660171  0.622331
Season_Summer         0.669553  0.762711  0.714616
Season_Winter         0.706533  0.793116  0.748574
```

```
Parameter1_Dir_N      0.728837  0.843042  0.783862
```

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 investgación y por qué? ¿Qué variables resultaron ser robustas a la especificación?

R: Existen diferencias entre los resultados mayormente entre el MCO y los otros modelos, y esto sucede debido a que el MCO no es tan bueno explicando variables binarias (Indicadora de falla en este caso), por lo cual el logit y el probit son más especificos a la hora de realizar mediciones. El que resulto mejor estimador para las variables es el modelo logit, ya que es el que explica el mayor porcentaje de datos (un 32.96% de estos), y las variables de interes terminaron siendo: La temperatura mínima y máxima, el Parameter4 a las 9am, la dirección del viento en "West" y "South" y las estaciones del año.

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.

R: Se eliminaron las variables dummies para el análisis de Poisson, según el R cuadrado se logro una estimación practicamente perfecta (Pseudo R-squ.(CS):0.9999), se estimo un número base de fallas mensuales de 33.48, un aumento en una unidad de las siguientes variables significaria lo siguiente para la cantidad de fallas: -La temperatura mínima: 0.1375 fallas más al mes -La temperatura máxima: 0.1344 fallas menos fallas al mes -Parameter3_3pm: 0.1044 fallas más al mes -Parameter4_3pm: 0.7000 fallas más al mes -Parameter6_9am_Bin: 0.9659 fallas menos al mes si es que existen datos medibles en el parametro

[22]:
```
df_filtrado = df[~((df['Date'].dt.year == 2008) | (df['Date'].dt.year == 2017)|␣
↪(df['Date'].dt.year == 2007))].copy()
```

[23]:
```
df_filtrado = df_filtrado.drop(columns=['Evaporation', 'Electricity'])
```

[24]:
```
# 1. Establecer la columna 'Date' como índice
df_filtrado = df_filtrado.set_index('Date')
# 2. Guardamos las fallas por mes
Month_failures = df_filtrado.resample('M')['Failure_today'].sum().
↪rename('Month_failures')
# 3. Resamplear por mes ('M') y calcular la media
df_promedios_mensuales = df_filtrado.resample('M').mean()
df_promedios_mensuales
```

[24]:

Date	Min_Temp	Max_Temp	Leakage	Parameter1_Speed	Parameter3_9am	\
2009-01-31	17.284982	30.417446	0.312276	45.338129	15.934353	
2009-02-28	17.687712	29.236364	0.494175	43.653347	15.652348	
2009-03-31	15.636965	26.985847	0.366321	40.877095	14.691806	
2009-04-30	12.818879	23.408879	0.513873	39.967290	15.025234	
2009-05-31	10.324079	20.369027	0.443316	37.110482	14.423041	

```
…                …          …          …                …                …
2016-08-31   8.135900  18.115219  0.609507         39.590868        14.658908
2016-09-30   9.867712  19.173063  0.743319         40.232472        15.367159
2016-10-31  10.773516  22.074225  0.512511         46.434012        17.712135
2016-11-30  13.243486  25.714954  0.321205         42.301835        15.206422
2016-12-31  15.980490  28.171442  0.415832         43.333648        15.428841

            Parameter3_3pm  Parameter4_9am  Parameter4_3pm  Parameter5_9am  \
Date
2009-01-31       21.382194       -0.510110       -0.476173     1013.302878
2009-02-28       20.935065       -0.194426       -0.232030     1013.314985
2009-03-31       19.749534       -0.011721       -0.199529     1016.438361
2009-04-30       19.053271        0.011591       -0.027310     1018.444019
2009-05-31       17.848914        0.232282        0.136787     1022.030784
…                      …               …               …               …
2016-08-31       18.843330        0.292866        0.229614     1020.217189
2016-09-30       19.283210        0.227262        0.397936     1016.084686
2016-10-31       22.765279       -0.287072       -0.083734     1015.570771
2016-11-30       20.883486       -0.385791       -0.184130     1014.096789
2016-12-31       20.726673       -0.322092       -0.191005     1012.901131

            Parameter5_3pm  …  Location_39  Location_40  Location_41  \
Date                        …
2009-01-31     1011.060072  …     0.026978     0.026978     0.025180
2009-02-28     1011.203197  …     0.024975     0.025974     0.021978
2009-03-31     1014.295717  …     0.024209     0.028864     0.021415
2009-04-30     1015.915981  …     0.023364     0.026168     0.025234
2009-05-31     1019.601794  …     0.027384     0.027384     0.020774
…                     …  …            …            …            …
2016-08-31     1017.685497  …     0.025962     0.026858     0.017905
2016-09-30     1013.461070  …     0.027675     0.024908     0.023063
2016-10-31     1013.375908  …     0.027458     0.023915     0.020372
2016-11-30     1011.819541  …     0.027523     0.026606     0.025688
2016-12-31     1010.723374  …     0.029218     0.025448     0.027333

            Location_43  Location_44  Location_45  Location_46  Location_47  \
Date
2009-01-31     0.027878     0.026978     0.027878     0.026978     0.025180
2009-02-28     0.027972     0.024975     0.026973     0.027972     0.024975
2009-03-31     0.027933     0.026071     0.025140     0.025140     0.022346
2009-04-30     0.026168     0.025234     0.025234     0.027103     0.027103
2009-05-31     0.026440     0.021719     0.025496     0.025496     0.021719
…                 …            …            …            …            …
2016-08-31     0.027753     0.020591     0.025067     0.027753     0.023277
2016-09-30     0.025830     0.023063     0.023985     0.011993     0.024908
2016-10-31     0.026572     0.022143     0.024801     0.019486     0.025686
2016-11-30     0.022018     0.021101     0.026606     0.023853     0.024771
```

```
2016-12-31     0.022620     0.029218     0.020735     0.003770     0.029218

            Location_48  Location_49
Date
2009-01-31     0.027878     0.027878
2009-02-28     0.027972     0.027972
2009-03-31     0.026071     0.027002
2009-04-30     0.026168     0.027103
2009-05-31     0.026440     0.029273
…                 …            …
2016-08-31     0.026858     0.027753
2016-09-30     0.027675     0.027675
2016-10-31     0.027458     0.026572
2016-11-30     0.026606     0.027523
2016-12-31     0.026390     0.018850

[96 rows x 69 columns]
```

[25]:
```
Month_failures
```

[25]:
```
Date
2009-01-31    141
2009-02-28    192
2009-03-31    195
2009-04-30    246
2009-05-31    209
              …
2016-08-31    330
2016-09-30    378
2016-10-31    298
2016-11-30    158
2016-12-31    207
Freq: ME, Name: Month_failures, Length: 96, dtype: int64
```

[26]:
```
#Eliminamos los meses sin datos:
df_promedios_mensuales=df_promedios_mensuales.dropna()
```

[27]:
```
df_promedios_mensuales
```

[27]:
```
             Min_Temp   Max_Temp   Leakage  Parameter1_Speed  Parameter3_9am  \
Date
2009-01-31  17.284982  30.417446  0.312276         45.338129       15.934353
2009-02-28  17.687712  29.236364  0.494175         43.653347       15.652348
2009-03-31  15.636965  26.985847  0.366321         40.877095       14.691806
2009-04-30  12.818879  23.408879  0.513873         39.967290       15.025234
2009-05-31  10.324079  20.369027  0.443316         37.110482       14.423041
…                 …          …         …                 …              …
```

```
2016-08-31   8.135900  18.115219  0.609507         39.590868       14.658908
2016-09-30   9.867712  19.173063  0.743319         40.232472       15.367159
2016-10-31  10.773516  22.074225  0.512511         46.434012       17.712135
2016-11-30  13.243486  25.714954  0.321205         42.301835       15.206422
2016-12-31  15.980490  28.171442  0.415832         43.333648       15.428841

            Parameter3_3pm  Parameter4_9am  Parameter4_3pm  Parameter5_9am  \
Date
2009-01-31       21.382194       -0.510110       -0.476173     1013.302878
2009-02-28       20.935065       -0.194426       -0.232030     1013.314985
2009-03-31       19.749534       -0.011721       -0.199529     1016.438361
2009-04-30       19.053271        0.011591       -0.027310     1018.444019
2009-05-31       17.848914        0.232282        0.136787     1022.030784
…                      …               …               …               …
2016-08-31       18.843330        0.292866        0.229614     1020.217189
2016-09-30       19.283210        0.227262        0.397936     1016.084686
2016-10-31       22.765279       -0.287072       -0.083734     1015.570771
2016-11-30       20.883486       -0.385791       -0.184130     1014.096789
2016-12-31       20.726673       -0.322092       -0.191005     1012.901131

            Parameter5_3pm  …  Location_39  Location_40  Location_41  \
Date                        …
2009-01-31     1011.060072  …     0.026978     0.026978     0.025180
2009-02-28     1011.203197  …     0.024975     0.025974     0.021978
2009-03-31     1014.295717  …     0.024209     0.028864     0.021415
2009-04-30     1015.915981  …     0.023364     0.026168     0.025234
2009-05-31     1019.601794  …     0.027384     0.027384     0.020774
…                     …  …            …            …            …
2016-08-31     1017.685497  …     0.025962     0.026858     0.017905
2016-09-30     1013.461070  …     0.027675     0.024908     0.023063
2016-10-31     1013.375908  …     0.027458     0.023915     0.020372
2016-11-30     1011.819541  …     0.027523     0.026606     0.025688
2016-12-31     1010.723374  …     0.029218     0.025448     0.027333

            Location_43  Location_44  Location_45  Location_46  Location_47  \
Date
2009-01-31     0.027878     0.026978     0.027878     0.026978     0.025180
2009-02-28     0.027972     0.024975     0.026973     0.027972     0.024975
2009-03-31     0.027933     0.026071     0.025140     0.025140     0.022346
2009-04-30     0.026168     0.025234     0.025234     0.027103     0.027103
2009-05-31     0.026440     0.021719     0.025496     0.025496     0.021719
…                 …            …            …            …            …
2016-08-31     0.027753     0.020591     0.025067     0.027753     0.023277
2016-09-30     0.025830     0.023063     0.023985     0.011993     0.024908
2016-10-31     0.026572     0.022143     0.024801     0.019486     0.025686
2016-11-30     0.022018     0.021101     0.026606     0.023853     0.024771
2016-12-31     0.022620     0.029218     0.020735     0.003770     0.029218
```

```
            Location_48  Location_49
Date
2009-01-31     0.027878     0.027878
2009-02-28     0.027972     0.027972
2009-03-31     0.026071     0.027002
2009-04-30     0.026168     0.027103
2009-05-31     0.026440     0.029273
…                 …            …
2016-08-31     0.026858     0.027753
2016-09-30     0.027675     0.027675
2016-10-31     0.027458     0.026572
2016-11-30     0.026606     0.027523
2016-12-31     0.026390     0.018850

[93 rows x 69 columns]
```

[28]:
```
#Eliminamos las variables categoricas
prefijo_a_eliminar = 'Season_'  # Ejemplo: eliminar todas las dummies que␣
 ↪comiencen con 'Season_'
columnas_a_mantener = [col for col in df_promedios_mensuales.columns if not col.
 ↪startswith(prefijo_a_eliminar)]
df_promedios_mensuales = df_promedios_mensuales[columnas_a_mantener]

prefijo_a_eliminar = 'Parameter1_Dir_'  # Ejemplo: eliminar todas las dummies␣
 ↪que comiencen con 'Season_'
columnas_a_mantener = [col for col in df_promedios_mensuales.columns if not col.
 ↪startswith(prefijo_a_eliminar)]
df_promedios_mensuales = df_promedios_mensuales[columnas_a_mantener]

prefijo_a_eliminar = 'Parameter2_9am_'  # Ejemplo: eliminar todas las dummies␣
 ↪que comiencen con 'Season_'
columnas_a_mantener = [col for col in df_promedios_mensuales.columns if not col.
 ↪startswith(prefijo_a_eliminar)]
df_promedios_mensuales = df_promedios_mensuales[columnas_a_mantener]

prefijo_a_eliminar = 'Parameter2_3pm_'  # Ejemplo: eliminar todas las dummies␣
 ↪que comiencen con 'Season_'
columnas_a_mantener = [col for col in df_promedios_mensuales.columns if not col.
 ↪startswith(prefijo_a_eliminar)]
df_promedios_mensuales = df_promedios_mensuales[columnas_a_mantener]

prefijo_a_eliminar = 'Location_'  # Ejemplo: eliminar todas las dummies que␣
 ↪comiencen con 'Season_'
columnas_a_mantener = [col for col in df_promedios_mensuales.columns if not col.
 ↪startswith(prefijo_a_eliminar)]
```

```
df_promedios_mensuales = df_promedios_mensuales[columnas_a_mantener]
```

[29]:
```
df_resultado = pd.concat([Month_failures, df_promedios_mensuales], axis=1)
```

[30]:
```
#Volvemos a aplicar un dropna() para asegurarnos de la limpieza de los datos
df_resultado=df_resultado.dropna()
```

[31]:
```
y = df_resultado['Month_failures']
X2=df_resultado.drop(['Month_failures','Failure_today','Leakage',␣
 ↪'Parameter5_3pm','Parameter7_3pm','Parameter7_9am','Parameter6_3pm_Bin'],␣
 ↪axis=1)
X2=sm.add_constant(X2)
X2 = X2.astype(float)
poisson=sm.GLM(y,X2,family=sm.families.Poisson()).fit()
print(poisson.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:         Month_failures   No. Observations:                   93
Model:                            GLM   Df Residuals:                       80
Model Family:                 Poisson   Df Model:                           12
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -457.57
Date:                Thu, 24 Apr 2025   Deviance:                       234.87
Time:                        21:55:56   Pearson chi2:                     233.
No. Iterations:                     4   Pseudo R-squ. (CS):             0.9999
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
const                33.4812     13.226      2.531      0.011       7.558
59.404
Min_Temp              0.1375      0.034      4.074      0.000       0.071
0.204
Max_Temp             -0.1344      0.034     -3.942      0.000      -0.201
-0.068
Parameter1_Speed     -0.0128      0.012     -1.073      0.283      -0.036
0.011
Parameter3_9am       -0.0310      0.026     -1.217      0.224      -0.081
0.019
Parameter3_3pm        0.1044      0.021      4.930      0.000       0.063
0.146
Parameter4_9am       -0.1574      0.125     -1.262      0.207      -0.402
0.087
Parameter4_3pm        0.7000      0.134      5.236      0.000       0.438
```

```
0.962
Parameter5_9am        -0.0083      0.005     -1.625      0.104      -0.018
0.002
Parameter6_9am_Bin    -0.9659      0.303     -3.192      0.001      -1.559
-0.373
EvaporationBin        -0.1135      0.319     -0.356      0.722      -0.738
0.511
ElectricityBin         0.1541      0.264      0.583      0.560      -0.364
0.672
Year                  -0.0092      0.007     -1.400      0.161      -0.022
0.004
==============================================================================
======
```

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

R: El valor de alpha obtenido es de 1.0062192597829778, lo cual nos indica una posible sobredispersión en los datos.

[32]:
```
predictions = poisson.predict(X2)
print(predictions)
```

```
Date
2009-01-31    155.392452
2009-02-28    211.882443
2009-03-31    197.420731
2009-04-30    226.380931
2009-05-31    230.264015
                 …
2016-08-31    278.677377
2016-09-30    359.267860
2016-10-31    281.347064
2016-11-30    216.433250
2016-12-31    221.732798
Length: 93, dtype: float64
```

[33]:
```
aux=((y-predictions)**2-predictions)/predictions
auxr=sm.OLS(aux,predictions).fit()
print(auxr.summary())
```

```
                                 OLS Regression Results
==============================================================================
=======
Dep. Variable:                      y   R-squared (uncentered):
0.203
Model:                            OLS   Adj. R-squared (uncentered):
0.195
Method:                 Least Squares   F-statistic:
23.46
```

```
Date:                Thu, 24 Apr 2025   Prob (F-statistic):
5.14e-06
Time:                        21:55:56   Log-Likelihood:
-236.08
No. Observations:                  93   AIC:
474.2
Df Residuals:                      92   BIC:
476.7
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.0062      0.001      4.844      0.000       0.004       0.009
==============================================================================
Omnibus:                       45.785   Durbin-Watson:                   2.088
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              106.506
Skew:                           1.881   Prob(JB):                     7.46e-24
Kurtosis:                       6.652   Cond. No.                         1.00
==============================================================================

Notes:
[1] R² 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.
```

[34]:
```
alfa=np.exp(0.0062)
print(alfa)
```

```
1.0062192597829778
```

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.

R: El modelo da como resultado que las variables que tenemos no explican practicamente nada los datos, teniendo un R cuadrado de casi 0 (Pseudo R-squ.(CS):0.03762) y con valores p mayores a 0.739, por lo cual no hay variables a rescatar.

[35]:
```
negbin = sm.GLM(y, X2, family=sm.families.NegativeBinomial(alpha=1.
↪0062192597829778)).fit()
print(negbin.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:         Month_failures   No. Observations:                   93
Model:                            GLM   Df Residuals:                       80
Model Family:        NegativeBinomial   Df Model:                           12
Link Function:                    Log   Scale:                          1.0000
```

```
Method:                           IRLS   Log-Likelihood:                -603.29
Date:                Thu, 24 Apr 2025   Deviance:                       0.96419
Time:                        21:55:57   Pearson chi2:                     0.945
No. Iterations:                     5   Pseudo R-squ. (CS):            0.03762
Covariance Type:            nonrobust
==============================================================================
======
                         coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
------
const                 30.0078    207.369      0.145      0.885    -376.428
436.443
Min_Temp               0.1614      0.528      0.306      0.760      -0.874
1.197
Max_Temp              -0.1575      0.535     -0.295      0.768      -1.205
0.890
Parameter1_Speed      -0.0061      0.188     -0.033      0.974      -0.374
0.362
Parameter3_9am        -0.0437      0.400     -0.109      0.913      -0.827
0.740
Parameter3_3pm         0.0980      0.331      0.296      0.767      -0.550
0.746
Parameter4_9am        -0.2329      1.949     -0.120      0.905      -4.052
3.586
Parameter4_3pm         0.7090      2.126      0.333      0.739      -3.458
4.876
Parameter5_9am        -0.0049      0.080     -0.061      0.951      -0.162
0.153
Parameter6_9am_Bin    -1.0777      4.828     -0.223      0.823     -10.540
8.385
EvaporationBin        -0.1873      5.131     -0.037      0.971     -10.244
9.870
ElectricityBin         0.2738      4.385      0.062      0.950      -8.320
8.867
Year                  -0.0090      0.102     -0.089      0.929      -0.209
0.191
==============================================================================
======
```

[36]:
```
X2["Predictions"] = negbin.predict(X2).astype(float)
print(X2.Predictions)
```

```
Date
2009-01-31    152.519627
2009-02-28    210.092124
2009-03-31    195.928544
2009-04-30    228.341360
```

```
2009-05-31    232.350027
                 …
2016-08-31    277.530433
2016-09-30    358.196594
2016-10-31    276.458187
2016-11-30    211.116893
2016-12-31    218.300866
Name: Predictions, Length: 93, dtype: float64
```

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 investgación y por qué? ¿Qué variables resultaron ser robustas a la especificación?

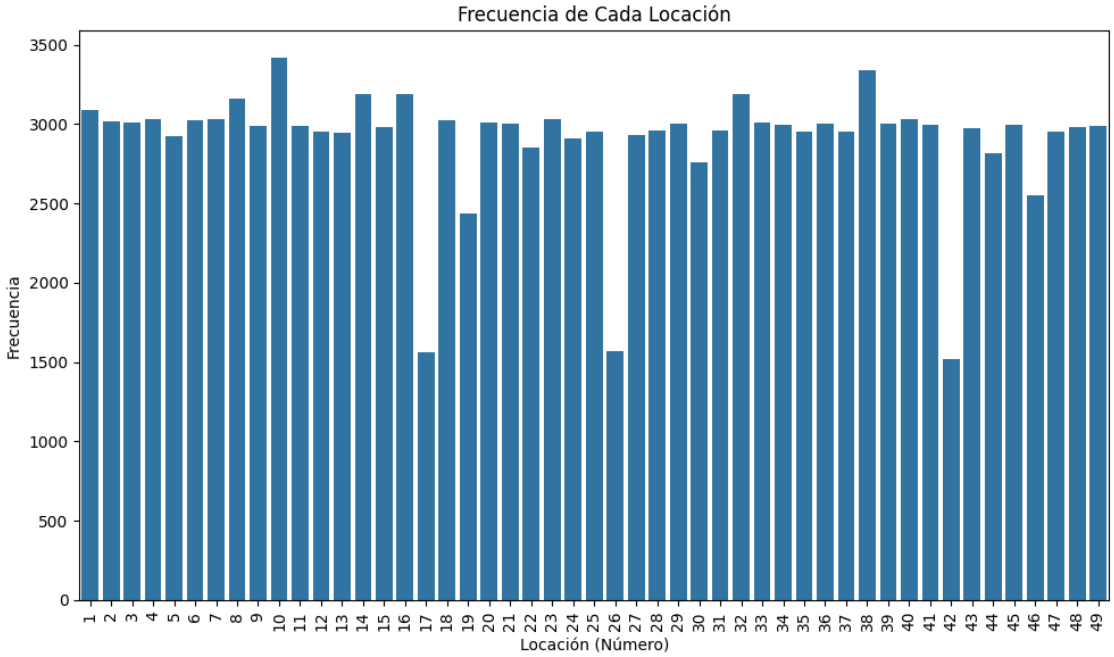
R: En Poisson se obtienen resultados completamente opuestos que en un modelo de Binomial Negativa, el primero explicando por completo varias variables, y el segundo no siendo capaz de explicar 1, debido a que existe cierto nivel de sobredispersión, es mas recomendable ocupar Binomial Negativa, pero al darnos resultados tan poco conluyentes, en lo personal investigaría las variables que nos arrojo Poisson, ya que es lo mejor que tenemos para poder sacar algún tipo de conclusión, las cuales fueron: la temperatura mínima, la temperatura máxima, Parameter3_3pm, Parameter4_3pm y Parameter6_9am_Bin

Anexos

[37]:
```
df_original = pd.read_csv('../../data/machine_failure_data.csv', delimiter=",",␣
↪decimal='.')
```

Anexo 1: Gráfico de freuencias de cada locación

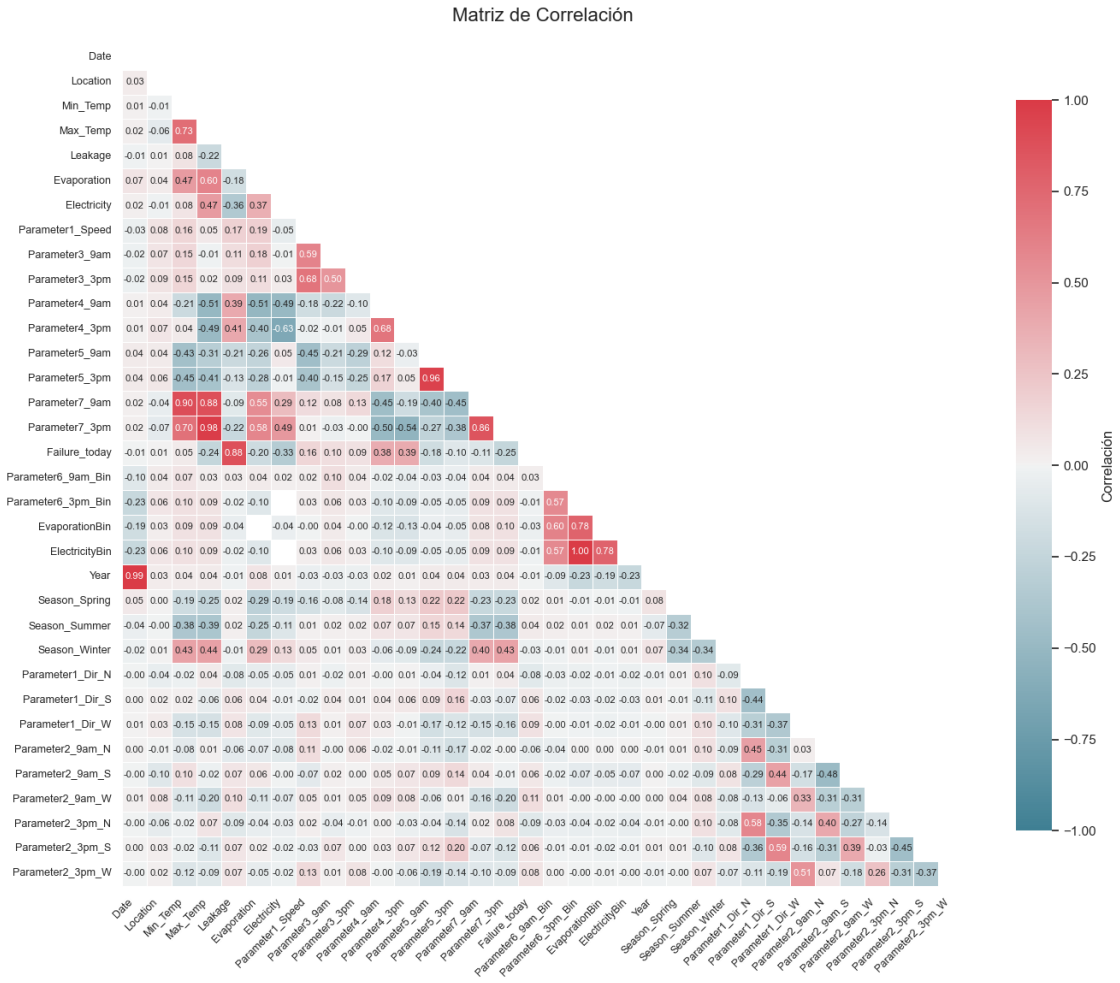
[38]:
```
plt.figure(figsize=(10, 6))
sns.countplot(x='Location', data=df_original)
plt.xlabel('Locación (Número)')
plt.ylabel('Frecuencia')
plt.title('Frecuencia de Cada Locación')
plt.xticks(rotation=90) # Rota las etiquetas del eje x si hay muchas locaciones
plt.tight_layout() # Ajusta el diseño para evitar que las etiquetas se␣
↪superpongan
plt.show()
```



Anexo 2: Matriz de correlación

[39]:
```
# Calcula la matriz de correlación
corr = dfcorrelaciones.corr()

# Máscara para la mitad superior
mask = np.triu(np.ones_like(corr, dtype=bool))

# Estilo de seaborn
sns.set(style="white")

# Tamaño del gráfico
f, ax = plt.subplots(figsize=(14, 12))

# Paleta de colores mejorada
cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Mapa de calor
sns.heatmap(
    corr,
    mask=mask,
    cmap=cmap,
    vmax=1.0,
    vmin=-1.0,
    center=0,
    square=True,
```

```
    linewidths=0.5,
    cbar_kws={"shrink": 0.8, 'label': 'Correlación'},
    annot=True,  # Muestra los valores numéricos
    fmt=".2f",   # Dos decimales
    annot_kws={"size": 8}
)

# Rota las etiquetas del eje X
plt.xticks(rotation=45, ha='right', fontsize=9)
plt.yticks(fontsize=9)

# Título
plt.title("Matriz de Correlación", fontsize=16, pad=20)

# Mostrar el gráfico
plt.tight_layout()
plt.show()
```



Anexo 3: Grafico “plambda”

[40]:
```
df_resultado['plambda'] = poisson.mu
sns.histplot(data=df_resultado, x="plambda")
```

[40]:
```
<Axes: xlabel='plambda', ylabel='Count'>
```