Tarea1_Martina_Larafinal

April 22, 2024

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.stats.api as sms
import statsmodels.formula.api as smf
import sklearn
import scipy
from scipy.stats import nbinom

import seaborn as sns
import plotly.express as px
%matplotlib inline
print("Librerias cargadas.")
```

```
Librerias cargadas.
```

1 Cargar la base de datos *disease.csv* en el ambiente. Identifique los tipos de datos que se encuentran en la base.

[2]:
```
#Carga de la base de datos
df= pd.read_csv("disease.csv")
#Tipos de datos de la base
df.dtypes
```

[2]:
```
Glucose                        float64
Cholesterol                    float64
Hemoglobin                     float64
Platelets                      float64
White Blood Cells              float64
Red Blood Cells                float64
Hematocrit                     float64
Mean Corpuscular Volume        float64
Mean Corpuscular Hemoglobin    float64
Insulin                        float64
BMI                            float64
Systolic Blood Pressure        float64
```

```
Diastolic Blood Pressure       float64
Triglycerides                  float64
HbA1c                          float64
LDL Cholesterol                float64
HDL Cholesterol                float64
Heart Rate                     float64
Creatinine                     float64
C-reactive Protein             float64
Disease                          int64
dtype: object
```

1.1 Realice estadisticas descriptivas sobre las variables importantes (Hint: Revisar la distribuciones, datos faltantes, outliers, etc.) y limpie las variables cuando sea necesario.

[3]:
```
#Se ajustó la variable Disease, a 0 si el individuo esta sano y 1 si esta␣
 ↪enfermo
df['Disease2'] = df['Disease'].replace([1, 2, 3, 4], 1)
#Elimina la columna Disease antigua
df_new= df.drop("Disease", axis=1)
```

[4]:
```
#Estadistica descriptiva
df_new.describe()
```

[4]:
```
            Glucose  Cholesterol   Hemoglobin    Platelets  White Blood Cells  \
count  2351.000000  2351.000000  2351.000000  2351.000000        2351.000000
mean      0.362828     0.393648     0.586190     0.504027           0.511086
std       0.251889     0.239449     0.271498     0.303347           0.277270
min       0.010994     0.012139     0.003021     0.012594           0.010139
25%       0.129198     0.195818     0.346092     0.200865           0.259467
50%       0.351722     0.397083     0.609836     0.533962           0.527381
75%       0.582278     0.582178     0.791215     0.754841           0.743164
max       0.968460     0.905026     0.983306     0.999393           0.990786

       Red Blood Cells   Hematocrit  Mean Corpuscular Volume
count      2351.000000  2351.000000              2351.000000  \
mean          0.506590     0.507152                 0.492200
std           0.266565     0.285537                 0.275735
min           0.044565     0.011772                 0.046942
25%           0.263589     0.288132                 0.287532
50%           0.467431     0.493428                 0.453052
75%           0.743670     0.753657                 0.722293
max           1.000000     0.977520                 0.995263

       Mean Corpuscular Hemoglobin      Insulin  …  Systolic Blood Pressure
count                  2351.000000  2351.000000  …             2351.000000  \
mean                      0.484459     0.447062  …                0.381211
```

```
std                      0.315618     0.242861  …                0.232785
min                      0.000554     0.034129  …                0.005988
25%                      0.207938     0.219111  …                0.179951
50%                      0.420723     0.444806  …                0.359064
75%                      0.778160     0.654441  …                0.580903
max                      0.963235     0.966784  …                0.829100

       Diastolic Blood Pressure  Triglycerides        HbA1c  LDL Cholesterol  \
count               2351.000000    2351.000000  2351.000000      2351.000000
mean                   0.421708       0.374373     0.439112         0.421777
std                    0.248768       0.256981     0.263779         0.252124
min                    0.005579       0.005217     0.016256         0.033037
25%                    0.175469       0.184604     0.188750         0.217757
50%                    0.474378       0.317857     0.466375         0.413071
75%                    0.663382       0.572330     0.652514         0.604753
max                    0.934617       0.973679     0.950218         0.983826

       HDL Cholesterol   Heart Rate   Creatinine  C-reactive Protein
count      2351.000000  2351.000000  2351.000000         2351.000000  \
mean          0.546079     0.582255     0.425075            0.430308
std           0.269511     0.250915     0.229298            0.243034
min           0.039505     0.114550     0.021239            0.004867
25%           0.307132     0.339125     0.213026            0.196192
50%           0.512941     0.610860     0.417295            0.481601
75%           0.779378     0.800666     0.606719            0.631426
max           0.989411     0.996873     0.925924            0.797906

          Disease2
count  2351.000000
mean      0.763505
std       0.425020
min       0.000000
25%       1.000000
50%       1.000000
75%       1.000000
max       1.000000

[8 rows x 21 columns]
```

El dataset no presenta outliers ni datos faltantes

Se utilizó un criterio de *Matriz de correlacion* para seleccionar las variables

Notamos que las variables con mayor relacion respecto a la variable dependiente son: Glucose, Platelets, White Blood Cells, Red Blood Cells, Hematocrit, Mean Corpuscular Volume, Mean Corpuscular Hemoglobin, Triglycerides y Heart Rate

Además se utilizó un criterio grafico de densidad condicional a la variable dependiente para la seleccion de variables

Al ver la densidad de las variables independientes condicional a la variable dependiente las que presentan mayor relevancia son: Glucose, Platelets,Red Blood Cells, Mean Corpuscular Volume, BMI, Triglycerides,LDL Cholesterol,Heart Rate y Creatinine. Debido a que la densidad de cada variable independiente presenta alta varianza y una relacion notoria respecto a la variable dependiente (VER GRAFICOS ANEXO)

2 Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.

Se concluye que las variables que cumplen ambos criterios estan estrechamente relacionadas con la variable Disease2, las cuales son: Glucose, Platelets, Red Blood Cells, Mean Corpuscular Volume, Triglycerides y Heart Rate. Sin embargo se deciden incluir en el analisis las variables BMI, LDL Cholesterol y Creatinine, las cuales mediante el metodo de la matriz de correlaciones no presentan una relacion notoria con la variable Disease2, pero esto puede tener varias causas que pueden significar que si son importantes en el analisis pero estan siendo afectadas por otras variables presentes en los datos que afecten en la relacion directa con la variable dependiente, se incluyen porque graficamente si tienen densidades que pueden explicar a la variable dependiente (presentan alta dispersión, patrones consistentes y tendencia) Por lo tanto el total de variables a considerar son:

Glucose, Platelets, Red Blood Cells, Mean Corpuscular Volume, Triglycerides, Heart Rate, BMI, LDL Cholesterol y Creatinine

[5]:
```
X=df_new[['Glucose','Creatinine','Platelets','Red Blood Cells','Mean␣
↪Corpuscular Volume','BMI','Triglycerides','LDL Cholesterol','Heart Rate']]
y=df_new['Disease2']
```

2.1 Ejecute un modelo de probabilidad lineal (*MCO*) que permita explicar la probabilidad de que una persona tenga al menos una enfermedad.

[6]:
```
X=sm.add_constant(X)
model = sm.OLS(y, X)
results = model.fit(cov_type='HC0')
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Disease2   R-squared:                       0.292
Model:                            OLS   Adj. R-squared:                  0.290
Method:                 Least Squares   F-statistic:                     161.1
Date:                Mon, 22 Apr 2024   Prob (F-statistic):          1.75e-237
Time:                        23:51:00   Log-Likelihood:                -917.32
No. Observations:                2351   AIC:                             1855.
Df Residuals:                    2341   BIC:                             1912.
Df Model:                           9
Covariance Type:                  HC0
==============================================================================
==========
```

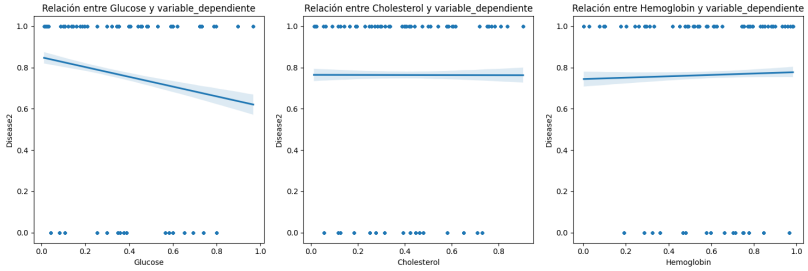
```
                             coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
-----------
const                      0.8289      0.036     23.267      0.000       0.759
0.899
Glucose                   -0.3152      0.032     -9.876      0.000      -0.378
-0.253
Creatinine                -0.1661      0.036     -4.634      0.000      -0.236
-0.096
Platelets                 -0.2811      0.023    -12.441      0.000      -0.325
-0.237
Red Blood Cells           -0.3943      0.029    -13.562      0.000      -0.451
-0.337
Mean Corpuscular Volume    0.5429      0.022     24.437      0.000       0.499
0.586
BMI                        0.0691      0.033      2.072      0.038       0.004
0.134
Triglycerides             -0.2411      0.032     -7.647      0.000      -0.303
-0.179
LDL Cholesterol            0.0885      0.029      3.045      0.002       0.032
0.145
Heart Rate                 0.3720      0.032     11.559      0.000       0.309
0.435
==============================================================================
Omnibus:                      237.066   Durbin-Watson:                   0.572
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              253.593
Skew:                          -0.757   Prob(JB):                    8.57e-56
Kurtosis:                       2.456   Cond. No.                        13.4
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
```

[7]:
```
residuals = results.resid
gq_test = sms.het_goldfeldquandt(residuals, X)

gq_test
```

[7]:
```
(1.0418935940075617e-29, 1.0, 'increasing')
```

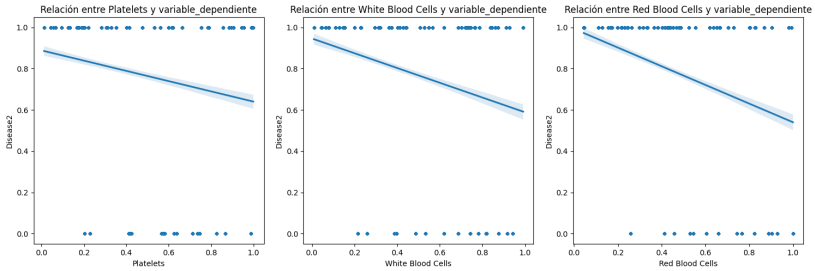
[8]:
```
# Calcula el número de columnas y filas para la cuadrícula
num_variables = len(df.columns) - 1  # El número de variables independientes
num_columnas = 3  # Puedes ajustar este valor según tus preferencias
num_filas = (num_variables + num_columnas - 1) // num_columnas

# Crea una figura y subplots
fig, axes = plt.subplots(num_filas, num_columnas, figsize=(15, num_filas * 5))
```

```
# Itera sobre cada variable independiente y crea un gráfico en el subplot␣
 ↪correspondiente
for i, variable in enumerate(df.columns[:-1]):
    fila = i // num_columnas
    columna = i % num_columnas
    sns.regplot(x=variable, y='Disease2', data=df, ax=axes[fila, columna],␣
 ↪scatter_kws={'s': 10})
    axes[fila, columna].set_title(f'Relación entre {variable} y␣
 ↪variable_dependiente')

# Ajusta el espaciado entre subgráficos
plt.tight_layout()

# Muestra el gráfico
plt.show()
```


Relación entre Glucose y variable_dependiente
Disease2
Glucose

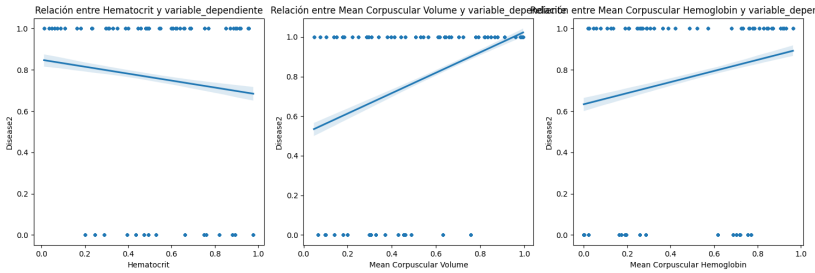
Relación entre Cholesterol y variable_dependiente
Disease2
Cholesterol

Relación entre Hemoglobin y variable_dependiente
Disease2
Hemoglobin


Relación entre Platelets y variable_dependiente
Disease2
Platelets

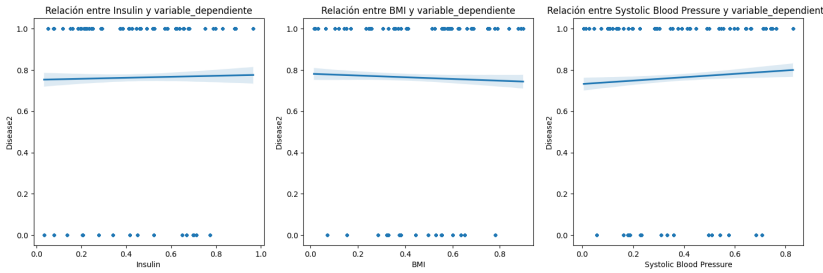
Relación entre White Blood Cells y variable_dependiente
Disease2
White Blood Cells

Relación entre Red Blood Cells y variable_dependiente
Disease2
Red Blood Cells


Relación entre Hematocrit y variable_dependiente
Disease2
Hematocrit

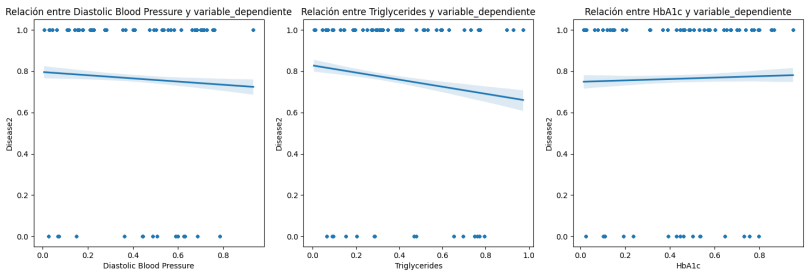
Relación entre Mean Corpuscular Volume y variable_dependiente
Disease2
Mean Corpuscular Volume

Relación entre Mean Corpuscular Hemoglobin y variable_dependiente
Disease2
Mean Corpuscular Hemoglobin


Relación entre Insulin y variable_dependiente
Disease2
Insulin

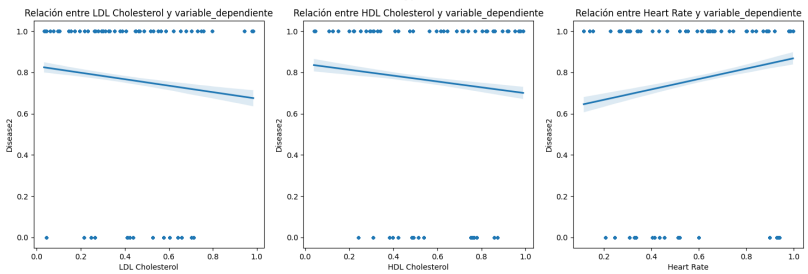
Relación entre BMI y variable_dependiente
Disease2
BMI

Relación entre Systolic Blood Pressure y variable_dependiente
Disease2
Systolic Blood Pressure


Relación entre Diastolic Blood Pressure y variable_dependiente
Disease2
Diastolic Blood Pressure

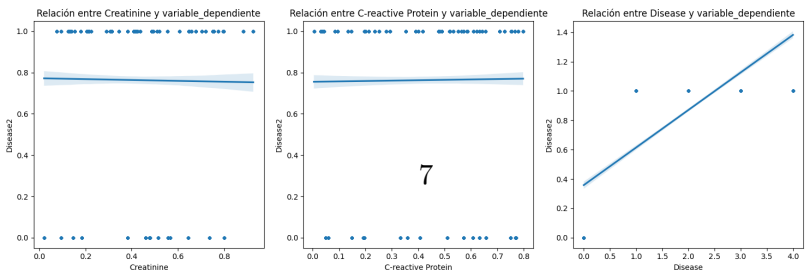
Relación entre Triglycerides y variable_dependiente
Disease2
Triglycerides

Relación entre HbA1c y variable_dependiente
Disease2
HbA1c


Relación entre LDL Cholesterol y variable_dependiente
Disease2
LDL Cholesterol

Relación entre HDL Cholesterol y variable_dependiente
Disease2
HDL Cholesterol

Relación entre Heart Rate y variable_dependiente
Disease2
Heart Rate


Relación entre Creatinine y variable_dependiente
Disease2
Creatinine

Relación entre C-reactive Protein y variable_dependiente
Disease2
C-reactive Protein

Relación entre Disease y variable_dependiente
Disease2
Disease

Se puede visualizar que la relacion entre las variables no es lineal por lo que el modelo no es apto para el analisis

2.1.1 Interpretacion de variables seleccionadas en *MCO*

Podemos notar que *todas las variables seleccionadas tienen un valor p menor a 0.05*, por lo que se consideran significativas en el modelo.Ademas con respecto a la bondad del ajuste el modelo es capaz de explicar un 29% de los datos. Variables con mayor coeficiente beta son:

Glucose:Si aumenta en una unidad "Glucose", la probabilidad de que una persona esté enferma disminuye en promedio en 31.52%

Heart Rate:Si aumenta en una unidad la frecuencia cardíaca "Heart Rate", la probabilidad de que una persona esté enferma aumenta en promedio en 37.20%.

Mean Corpuscular Volume:Si "Mean Corpuscular Volume" aumenta en una unidad, la probabilidad de que una persona esté enferma aumenta en promedio en 54.29 %

Red Blood Cells:Si "Red Blood Cells" aumenta en una unidad, la probabilidad de que una persona esté enferma disminuye en promedio 39.43% unidades.

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

[9]:
```
#probit
X=sm.add_constant(X)

model_probit = sm.Probit(y,X).fit()
print(model_probit.summary())

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

```
Optimization terminated successfully.
         Current function value: 0.353852
         Iterations 8
                          Probit Regression Results
==============================================================================
Dep. Variable:               Disease2   No. Observations:                 2351
Model:                         Probit   Df Residuals:                     2341
Method:                           MLE   Df Model:                            9
Date:                Mon, 22 Apr 2024   Pseudo R-squ.:                  0.3531
Time:                        23:51:04   Log-Likelihood:                -831.91
converged:                       True   LL-Null:                       -1286.0
Covariance Type:            nonrobust   LLR p-value:                1.052e-189
==============================================================================
===========
```

```
                              coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
-----------
const                         1.8094      0.256      7.074      0.000       1.308
2.311
Glucose                      -1.7192      0.170    -10.111      0.000      -2.052
-1.386
Creatinine                   -1.4310      0.172     -8.321      0.000      -1.768
-1.094
Platelets                    -1.9091      0.156    -12.273      0.000      -2.214
-1.604
Red Blood Cells              -2.2115      0.156    -14.175      0.000      -2.517
-1.906
Mean Corpuscular Volume       3.3466      0.213     15.681      0.000       2.928
3.765
BMI                           0.7614      0.175      4.356      0.000       0.419
1.104
Triglycerides                -1.5343      0.160     -9.612      0.000      -1.847
-1.221
LDL Cholesterol               0.3141      0.164      1.921      0.055      -0.006
0.635
Heart Rate                    1.9931      0.167     11.908      0.000       1.665
2.321
==============================================================================
===========
       Probit Marginal Effects
=====================================
Dep. Variable:               Disease2
Method:                          dydx
At:                           overall
==============================================================================
===========
                              dy/dx    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
-----------
Glucose                      -0.3391      0.031    -10.782      0.000      -0.401
-0.277
Creatinine                   -0.2822      0.032     -8.745      0.000      -0.345
-0.219
Platelets                    -0.3765      0.027    -14.083      0.000      -0.429
-0.324
Red Blood Cells              -0.4361      0.027    -16.204      0.000      -0.489
-0.383
Mean Corpuscular Volume       0.6600      0.034     19.476      0.000       0.594
0.726
BMI                           0.1502      0.034      4.407      0.000       0.083
```

```
0.217
Triglycerides               -0.3026      0.029    -10.283      0.000      -0.360
-0.245
LDL Cholesterol              0.0619      0.032      1.923      0.055      -0.001
0.125
Heart Rate                   0.3931      0.030     13.072      0.000       0.334
0.452
==============================================================================
==========
```

3.1 Interpretación variables seleccionadas *Modelo probit*

Aparentemente no deberia considerarse la variable LDL Cholesterol, debido a que el valor p de sus efectos marginales es mayor a 0.05, pero al ser una minima diferencia de 0.05 se decide considerar.El resto de las variables cumple con el criterio de significancia, siendo las variables mas influyentes en Disease2 y su interpretacion de los coeficientes de los efectos marginales de cada variable son:

Red Blood Cells: Un aumento de una unidad en el recuento de glóbulos rojos se asocia, en promedio, con una disminución de aproximadamente 43.61% en la probabilidad de que una persona esté enferma.

Heart Rate:Un aumento de una unidad en la frecuencia cardíaca se asocia, en promedio, con un aumento de aproximadamente 39.31% en la probabilidad de que una persona esté enferma.

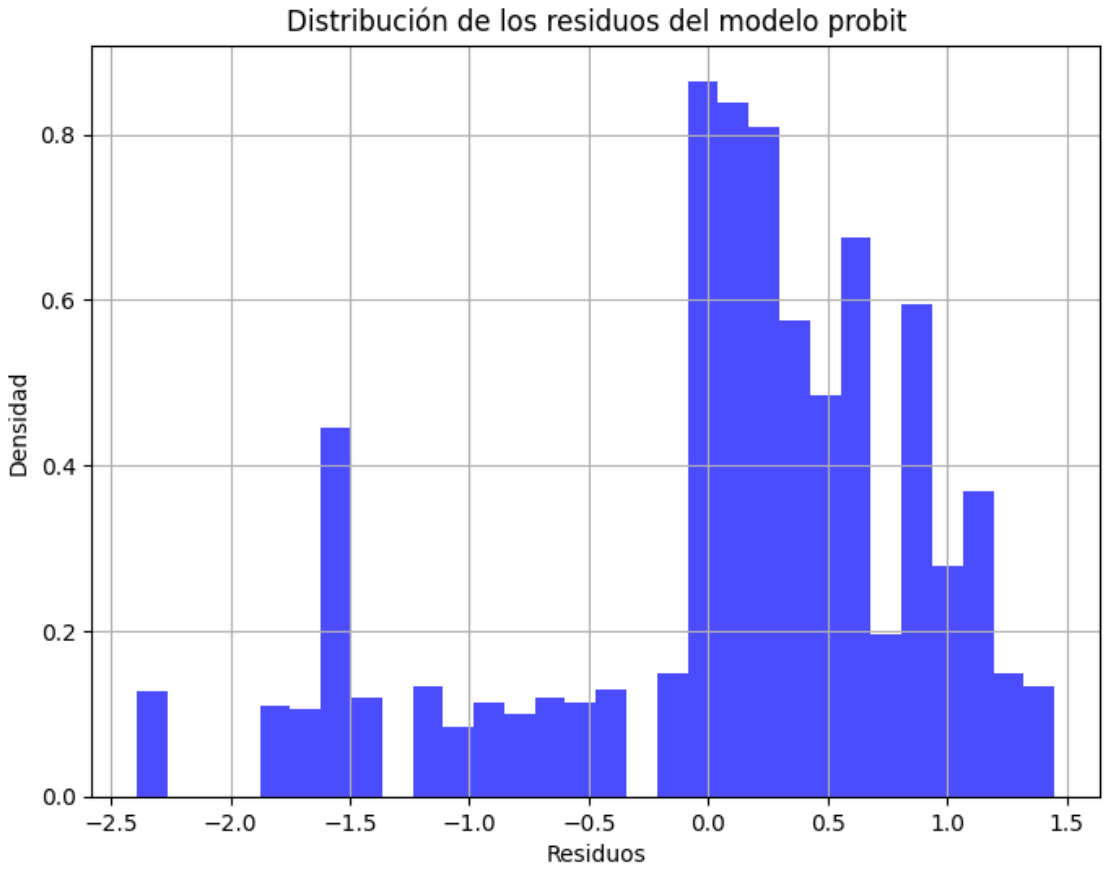
Mean Corpuscular Volume:Un aumento de una unidad en el volumen corpuscular se asocia, en promedio, con un aumento de aproximadamente 66% en la probabilidad de que una persona esté enferma.

Platelets: Un aumento de una unidad en el recuento de plaquetas se asocia, en promedio, con una disminución de aproximadamente 37.65% en la probabilidad de que una persona esté enferma.

[10]:
```
residuoP= model_probit.resid_dev
gq_testP = sms.het_goldfeldquandt(residuoP, X)
gq_testP
```

[10]:
```
(0.01893805524434349, 0.9999999999999999, 'increasing')
```

[11]:
```
# 3. Grafica la distribución de los residuos
plt.figure(figsize=(8, 6))
plt.hist(residuoP, bins=30, density=True, alpha=0.7, color='b')
plt.title('Distribución de los residuos del modelo probit')
plt.xlabel('Residuos')
plt.ylabel('Densidad')
plt.grid(True)
plt.show()
```



[12]:
```
from scipy.stats import shapiro

# Aplica el test de Shapiro-Wilk
estadistico, p_valor = shapiro(residuoP)

# Imprime el resultado
print("Estadístico de prueba:", estadistico)
print("Valor p:", p_valor)

# Comprueba el valor p para determinar si rechazamos la hipótesis nula
nivel_significancia = 0.05
if p_valor > nivel_significancia:
    print("No se rechaza la hipótesis nula. Los datos pueden provenir de una␣
 ↪distribución normal.")
else:
    print("Se rechaza la hipótesis nula. Los datos no provienen de una␣
 ↪distribución normal.")
```

```
Estadístico de prueba: 0.8988901972770691
Valor p: 7.814294923717713e-37
```

Se rechaza la hipótesis nula. Los datos no provienen de una distribución normal.

Se identifica que los errores no se distribuyen normal, por lo que no se cumple el supuesto y el modelo no seria apto para el analisis

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.

[13]:
```
#logit
X = sm.add_constant(X)

model_logit = sm.Logit(y,X).fit()
print(model_logit.summary())

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

```
Optimization terminated successfully.
         Current function value: 0.355056
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:               Disease2   No. Observations:                 2351
Model:                          Logit   Df Residuals:                     2341
Method:                           MLE   Df Model:                            9
Date:                Mon, 22 Apr 2024   Pseudo R-squ.:                  0.3509
Time:                        23:51:04   Log-Likelihood:                -834.74
converged:                       True   LL-Null:                       -1286.0
Covariance Type:            nonrobust   LLR p-value:                1.746e-188
==============================================================================
                                 coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
const                          3.0913      0.454      6.816      0.000       2.202
3.980
Glucose                       -3.1871      0.320     -9.948      0.000      -3.815
-2.559
Creatinine                    -2.4494      0.298     -8.213      0.000      -3.034
-1.865
Platelets                     -3.1054      0.261    -11.882      0.000      -3.618
-2.593
Red Blood Cells               -4.0675      0.299    -13.603      0.000      -4.654
-3.481
Mean Corpuscular Volume        5.6549      0.375     15.062      0.000       4.919
```

```
6.391
BMI                          1.3308      0.298      4.463      0.000       0.746
1.915
Triglycerides               -2.6465      0.277     -9.558      0.000      -3.189
-2.104
LDL Cholesterol              0.7032      0.280      2.514      0.012       0.155
1.251
Heart Rate                   3.6537      0.309     11.842      0.000       3.049
4.258
================================================================================
===========
        Logit Marginal Effects
=====================================
Dep. Variable:                Disease2
Method:                           dydx
At:                            overall
================================================================================
===========
                               dy/dx    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-----------
Glucose                     -0.3629      0.034    -10.766      0.000      -0.429
-0.297
Creatinine                  -0.2789      0.032     -8.633      0.000      -0.342
-0.216
Platelets                   -0.3536      0.026    -13.455      0.000      -0.405
-0.302
Red Blood Cells             -0.4631      0.029    -15.853      0.000      -0.520
-0.406
Mean Corpuscular Volume      0.6439      0.034     18.666      0.000       0.576
0.711
BMI                          0.1515      0.034      4.522      0.000       0.086
0.217
Triglycerides               -0.3013      0.029    -10.263      0.000      -0.359
-0.244
LDL Cholesterol              0.0801      0.032      2.529      0.011       0.018
0.142
Heart Rate                   0.4160      0.031     13.375      0.000       0.355
0.477
================================================================================
===========
```

[14]:
```
residuoL= model_logit.resid_dev
gq_testL = sms.het_goldfeldquandt(residuoL, X)
gq_testL
```

[14]: (0.019030101135156807, 0.9999999999999999, 'increasing')

[15]:
```
#Ratio OR
beta_mean_cv= 0.6469

# Calcula el ratio de probabilidades (OR)
OR = np.exp(beta_mean_cv)
print("Ratio de Probabilidades (OR) para la variable independiente:", OR)
```

```
Ratio de Probabilidades (OR) para la variable independiente: 1.9096118471140235
```

Como el valor del ratio es mayor a 1 indica que indica que la variable mean corpuscular volume está asociada con un aumento en la probabilidad de tener la enfermedad.

4.1 Interpretación variables seleccionadas *Modelo logit*

Todos los valores p son menores a 0.05 por lo que estadisticamente son significativos, Ademas el pseudo r cuadrado nos indica que el modelo explica aproximadamente un 35% de los datos. Siendo mas relevantes los coeficientes de los efectos marginales de las variables:

Mean Corpuscular Volume:Un aumento de una unidad en el volumen corpuscular medio aumenta la probabilidad de tener la enfermedad en aproximadamente 64.39%.

Platelets:Un aumento de una unidad en los niveles de plaquetas disminuye la probabilidad de tener la enfermedad en aproximadamente 35.36%

Heart Rate:Un aumento de una unidad en el ritmo cardíaco aumenta la probabilidad de tener la enfermedad en aproximadamente 41.60%

Red Blood Cells:Un aumento de una unidad en los niveles de glóbulos rojos disminuye la probabilidad de tener la enfermedad en aproximadamente 46.31%

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?

Los modelos logit y probit son mas adecuados para predecir comportamiento de variables dicotomicas, porque utilizan funciones de acumulación de distribución (la función logística para el Logit y la función de distribución normal acumulativa para el Probit) para modelar la relación entre las variables independientes y la probabilidad de la variable dependiente, cabe destacar que un modelo **MCO no es bueno explicando una variable dependiente de naturaleza binaria**,ya que asume una relación lineal entre las variables y no tiene en cuenta la naturaleza discreta de la variable dependiente.El mas adecuado en este caso es el LOGIT porque es mas adecuado modelando datos de ciencias de la salud y los resultados son acorde a ese criterio. En mi opinion las variables mas robustas a la especificacion son Mean Corpuscular Volume, Heart Rate y Red Blood Cells, pues estas tres variables presentan alto coeficiente beta en los tres modelos y su valor p es menor a 0.05 en todos, por lo que son capaces de explicar la probabilidad de que una persona este enferma.

6 Ejecute un modelo Poisson para explicar el numero de enfermedades que tiene una persona. Seleccione las variables dependientes a incluir en el modelo final e interprete su significado.

[16]:
```
df_2= df.drop("Disease2", axis=1)
```

No se consideran White Blood Cells, Red Blood Cells, Insulin y Heart Rate,debido a que no se encuentran patrones consistentes graficamente en comparacion a la variable dependiente

[17]:
```
xx= df_2[['Glucose', 'Cholesterol', 'Hemoglobin','Platelets','Hematocrit','Mean␣
 ↪Corpuscular Volume','Systolic Blood Pressure','Diastolic Blood␣
 ↪Pressure','Mean Corpuscular Hemoglobin','BMI','Triglycerides','HbA1c','LDL␣
 ↪Cholesterol','HDL Cholesterol','Creatinine', 'C-reactive Protein']]
yy= df_2['Disease']
```

[18]:
```
poisson=sm.GLM(yy,xx,family=sm.families.Poisson()).fit()
print(poisson.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                Disease   No. Observations:                 2351
Model:                            GLM   Df Residuals:                     2335
Model Family:                 Poisson   Df Model:                           15
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -3153.0
Date:                Mon, 22 Apr 2024   Deviance:                       1724.2
Time:                        23:51:05   Pearson chi2:                 1.31e+03
No. Iterations:                     5   Pseudo R-squ. (CS):             0.3663
Covariance Type:            nonrobust
==============================================================================
===============
                                 coef    std err          z      P>|z|
[0.025      0.975]
------------------------------------------------------------------------------
---------------
Glucose                       -0.6067      0.081     -7.495      0.000
-0.765      -0.448
Cholesterol                    0.7976      0.083      9.632      0.000
0.635       0.960
Hemoglobin                     1.0998      0.077     14.207      0.000
0.948       1.251
Platelets                     -1.3474      0.060    -22.442      0.000
-1.465      -1.230
Hematocrit                    -0.3518      0.069     -5.128      0.000
-0.486      -0.217
Mean Corpuscular Volume        0.9068      0.064     14.205      0.000
0.782       1.032
Systolic Blood Pressure       -0.4060      0.076     -5.360      0.000
```

```
-0.555      -0.258
Diastolic Blood Pressure       -0.5762      0.080     -7.225      0.000
-0.732      -0.420
Mean Corpuscular Hemoglobin     0.7138      0.058     12.397      0.000
0.601       0.827
BMI                             1.1123      0.092     12.126      0.000
0.932       1.292
Triglycerides                  -0.5128      0.072     -7.130      0.000
-0.654      -0.372
HbA1c                           0.2741      0.066      4.167      0.000
0.145       0.403
LDL Cholesterol                -0.9512      0.079    -12.097      0.000
-1.105      -0.797
HDL Cholesterol                -0.4753      0.059     -8.078      0.000
-0.591      -0.360
Creatinine                      0.3287      0.076      4.324      0.000
0.180       0.478
C-reactive Protein              0.3549      0.073      4.841      0.000
0.211       0.499
==============================================================================
==============
```

[19]:
```
print("fitted lambda")
print(poisson.mu)
```

```
fitted lambda
[0.80023608 0.66643208 1.40719757 … 3.41786189 3.1302448  3.41786189]
```

6.1 Interpretacion modelo *poisson*

Se concluye que todas las variables son estadisticamente significativas, cumpliendo el criterio de signficancia con un valor p inferior 0.05, las variables con mayor incidencia en que una persona tenga mayor cantidad de enfermedades. Por ejemplo si aumenta platelets la probabilidad de tener mas de alguna enfermedad disminuye o por ejemplo si aumenta Hemoglobin la probabilidad de tener mas de alguna enfermedad aumenta (ya que al tener una hemoglobina muy alta puede provocar coagulos sanguineos que generan ataques cardiovasculares o cardiacos, aumentando el riesgo de padecer enfermedades). Ademas el modelo es capaz de explicar aproximadamente un 36% de los datos

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

[20]:
```
#Test de sobre dispersion
aux=((yy-poisson.mu)**2-poisson.mu)/poisson.mu
auxr=sm.OLS(aux,poisson.mu).fit()
print(auxr.summary())
```

```
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:                Disease   R-squared (uncentered):
0.403
Model:                            OLS   Adj. R-squared (uncentered):
0.403
Method:                 Least Squares   F-statistic:
1589.
Date:                Mon, 22 Apr 2024   Prob (F-statistic):
7.54e-266
Time:                        23:51:05   Log-Likelihood:
-1842.9
No. Observations:                2351   AIC:
3688.
Df Residuals:                    2350   BIC:
3694.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1            -0.2396      0.006    -39.859      0.000      -0.251      -0.228
==============================================================================
Omnibus:                      455.446   Durbin-Watson:                   1.286
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              812.746
Skew:                           1.216   Prob(JB):                    3.27e-177
Kurtosis:                       4.543   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.
```

[21]:
```
#Posible valor optimo de alfa
print(np.exp(-0.2396))
```

```
0.7869425751496004
```

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

[22]:
```
negbin=sm.GLM(yy,xx,family=sm.families.NegativeBinomial(alpha=0.786)).fit()
print(negbin.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                 Disease   No. Observations:                 2351
Model:                             GLM   Df Residuals:                     2335
Model Family:        NegativeBinomial   Df Model:                           15
Link Function:                     Log   Scale:                          1.0000
Method:                           IRLS   Log-Likelihood:                -3705.1
Date:                 Mon, 22 Apr 2024   Deviance:                       1031.1
Time:                         23:51:05   Pearson chi2:                     675.
No. Iterations:                      10   Pseudo R-squ. (CS):             0.2007
Covariance Type:             nonrobust
==============================================================================
===============
                                 coef    std err          z      P>|z|
[0.025      0.975]
------------------------------------------------------------------------------
---------------
Glucose                       -0.5789      0.127     -4.572      0.000
-0.827      -0.331
Cholesterol                    0.8912      0.126      7.047      0.000
0.643       1.139
Hemoglobin                     1.0902      0.116      9.390      0.000
0.863       1.318
Platelets                     -1.5237      0.092    -16.485      0.000
-1.705      -1.343
Hematocrit                    -0.4745      0.102     -4.648      0.000
-0.675      -0.274
Mean Corpuscular Volume        1.1254      0.096     11.693      0.000
0.937       1.314
Systolic Blood Pressure       -0.4167      0.116     -3.600      0.000
-0.644      -0.190
Diastolic Blood Pressure      -0.6530      0.122     -5.347      0.000
-0.892      -0.414
Mean Corpuscular Hemoglobin    0.8148      0.090      9.081      0.000
0.639       0.991
BMI                            1.1593      0.135      8.615      0.000
0.896       1.423
Triglycerides                 -0.6403      0.110     -5.843      0.000
-0.855      -0.426
```

```
HbA1c                          0.3849      0.103      3.730      0.000
0.183       0.587
LDL Cholesterol               -0.9777      0.119     -8.212      0.000
-1.211      -0.744
HDL Cholesterol               -0.5675      0.091     -6.256      0.000
-0.745      -0.390
Creatinine                     0.3971      0.118      3.355      0.001
0.165       0.629
C-reactive Protein             0.3087      0.115      2.695      0.007
0.084       0.533
==============================================================================
===============
```

8.1 Interpretacion *Modelo Binomial negativa*

Se concluye que todas las variables son estadisticamente significativas, cumpliendo el criterio de signficancia con un valor p inferior 0.05 ademas, explica aproximadamente el 20% de la variabilidad en la variable dependiente. Las variables con mayor incidencia en que una persona tenga mayor cantidad de enfermedades son: Platelets y Mean Corpuscular Volume, por ejemplo al aumentar Platelets disminuye la probabilidad de que la persona contraiga una mayor cantidad de enfermedades o tambien al aumentar el volumen corpuscular medio la probabilidad de contraer mayor cantidad de enfermedades aumenta.

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?

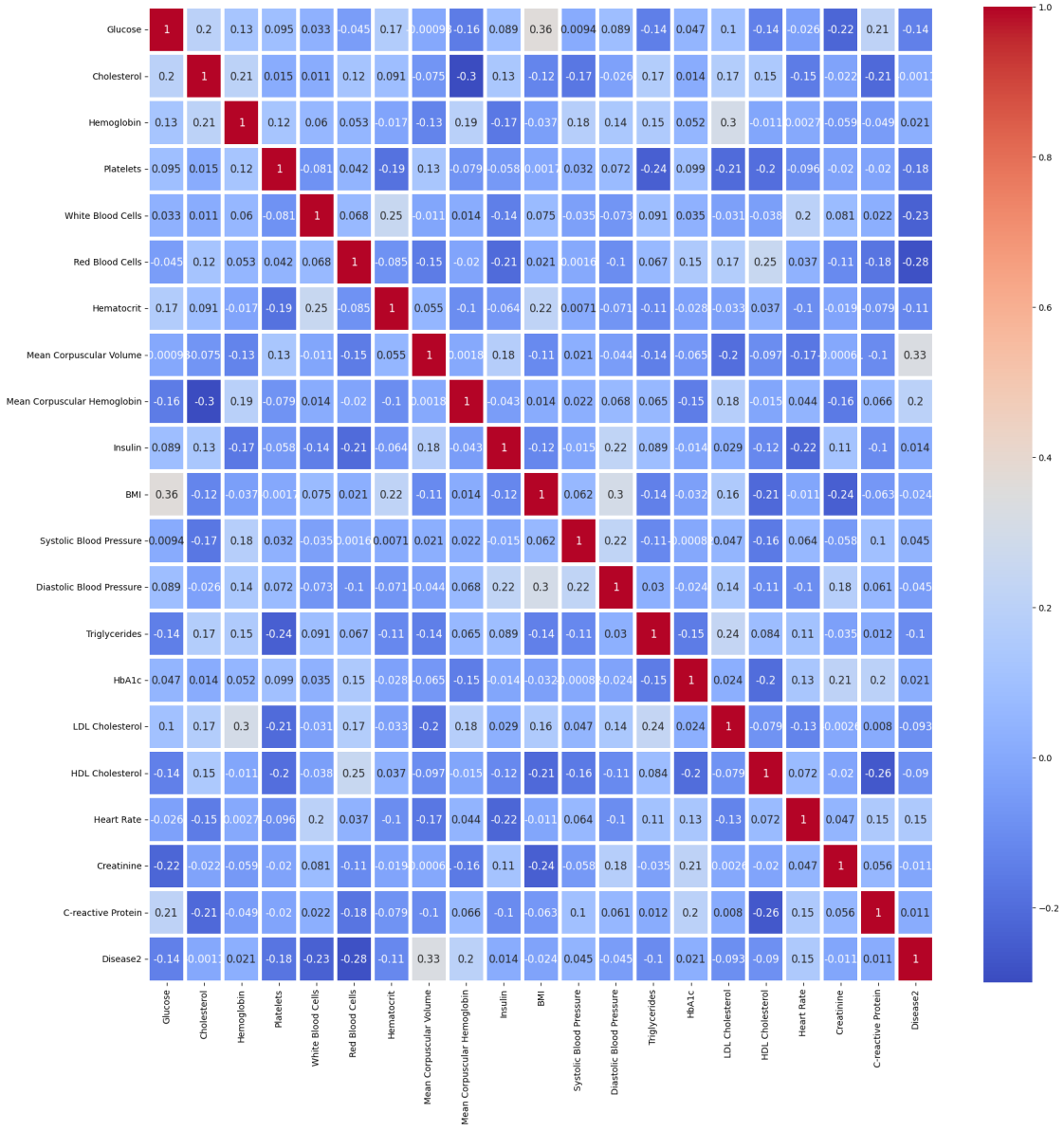
Podemos notar que el test de sobre dispersión nos entregó un valor negativo, lo que indica que el Modelo Binomial Negativa no es apto para realizar analisis con estos datos, ya que no se ajustan. En general los resultados fueron similares entre ambos modelos. En mi opinion el modelo de poisson es mas apto para el analisis debido a que no presentó sobredispersion.Ademas se puede destacar que las variables que resultaron ser robustas a la explicacion tales como: Platelets, Mean Corpuscular Volume, BMI y Hemoglobin, presentan de los coeficientes mas altos en los modelos, que son capaces de explicar bien a la variable dependiente.

10 ANEXOS

Criterio matriz de correlacion

[23]:
```
plt.figure(figsize=(20,20))
sns.heatmap(df_new.corr(), cmap='coolwarm', annot=True, linecolor='white',␣
 ↪linewidths=4, annot_kws={"fontsize":12})
```

[23]: <Axes: >



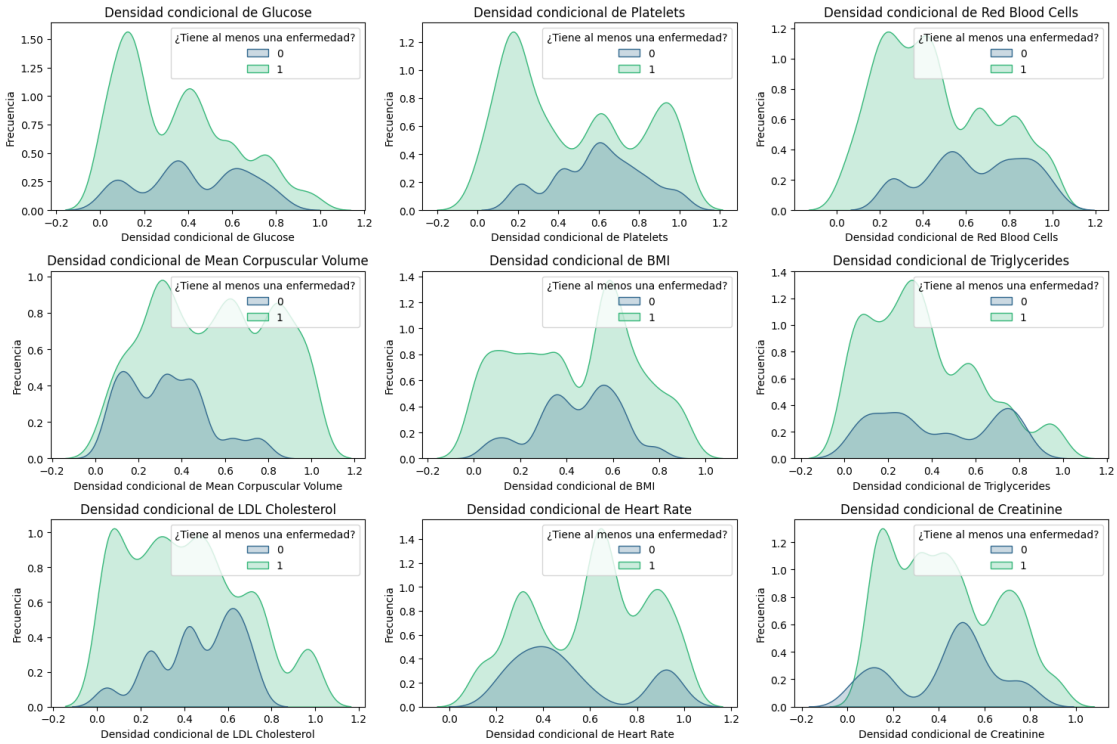
Criterio densidad condicional

Densidad condicional de variables elegidas

[24]:
```
# Definir el tamaño de la figura
plt.figure(figsize=(15, 10))

# Lista de columnas que deseas visualizar
var_elejidas=["Glucose", "Platelets", "Red Blood Cells", "Mean Corpuscular␣
 ↪Volume", "BMI", "Triglycerides", "LDL Cholesterol", "Heart Rate",␣
 ↪"Creatinine"]
```

```
# Número de filas y columnas en la cuadrícula de subplots
num_rows = 3
num_cols = 3

# Crear subplots
fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 10))

# Iterar sobre cada columna del DataFrame y asignarla a un subplot
for i, column in enumerate(var_elejidas):
    row = i // num_cols
    col = i % num_cols

    # Generar el histplot para la columna actual
    sns.kdeplot(data=df_new, x=column, hue='Disease2', fill=True,␣
 ↪palette='viridis', ax=axes[row, col])

    # Cambiar el título del "hue" en la leyenda
    axes[row, col].legend_.set_title('¿Tiene al menos una enfermedad?')

    # Agregar título y etiquetas
    axes[row, col].set_title(f'Densidad condicional de {column}')
    axes[row, col].set_xlabel(f'Densidad condicional de {column}')
    axes[row, col].set_ylabel('Frecuencia')


# Ajustar espacios entre subplots
plt.tight_layout()

# Mostrar el gráfico
plt.show()
```

```
<Figure size 1500x1000 with 0 Axes>
```

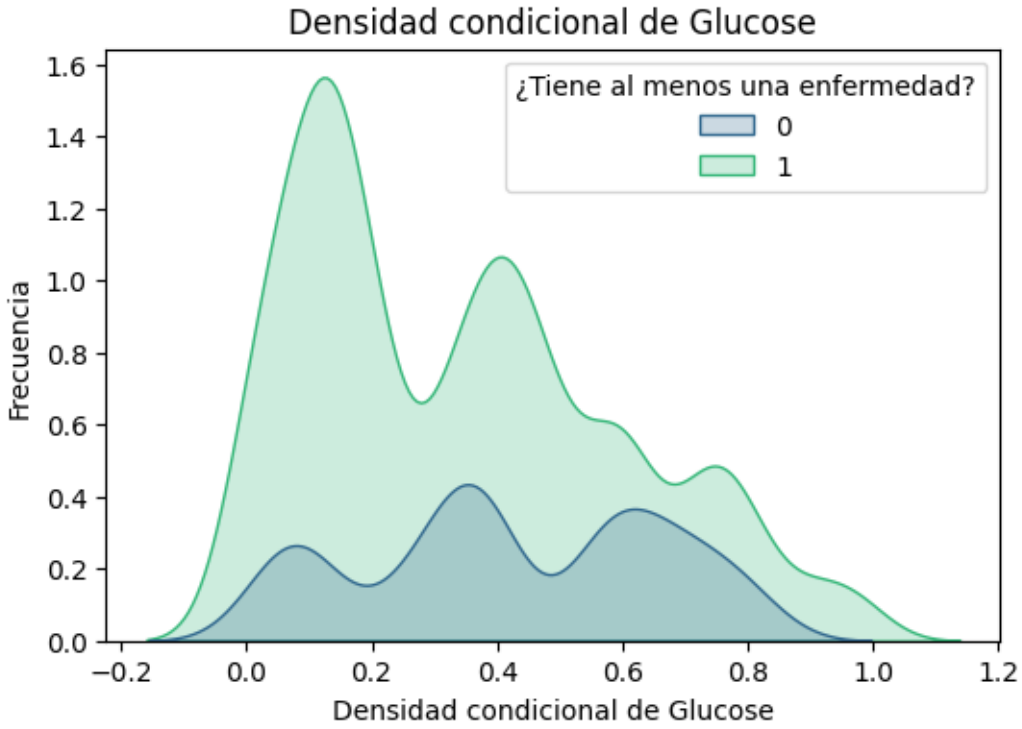


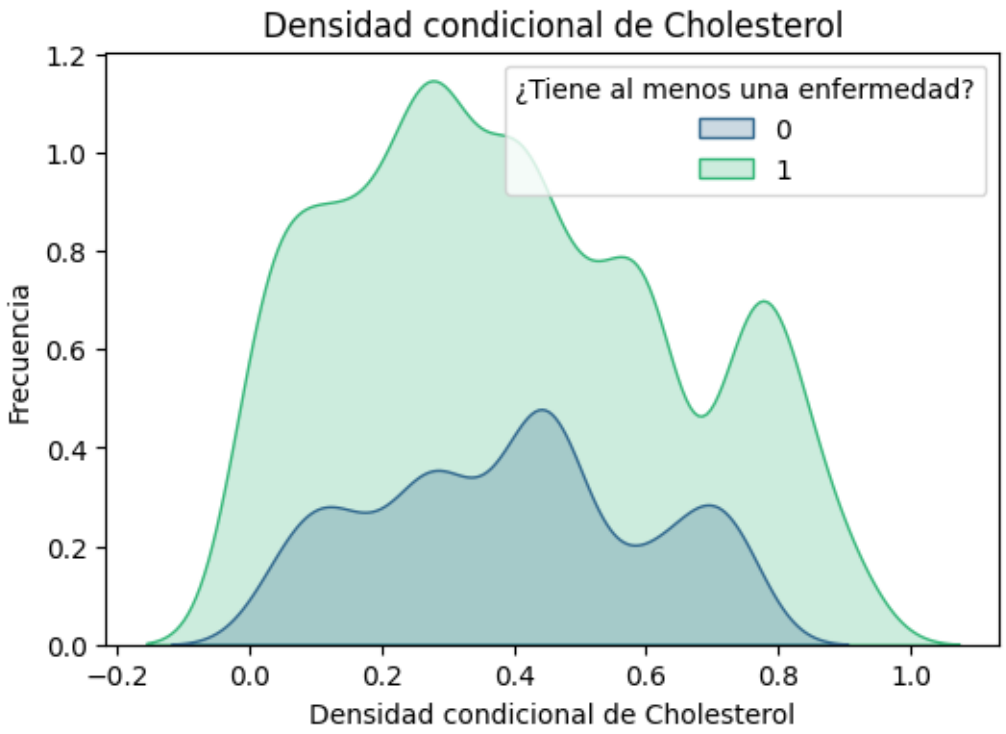
Densidad condicional de todas las variables

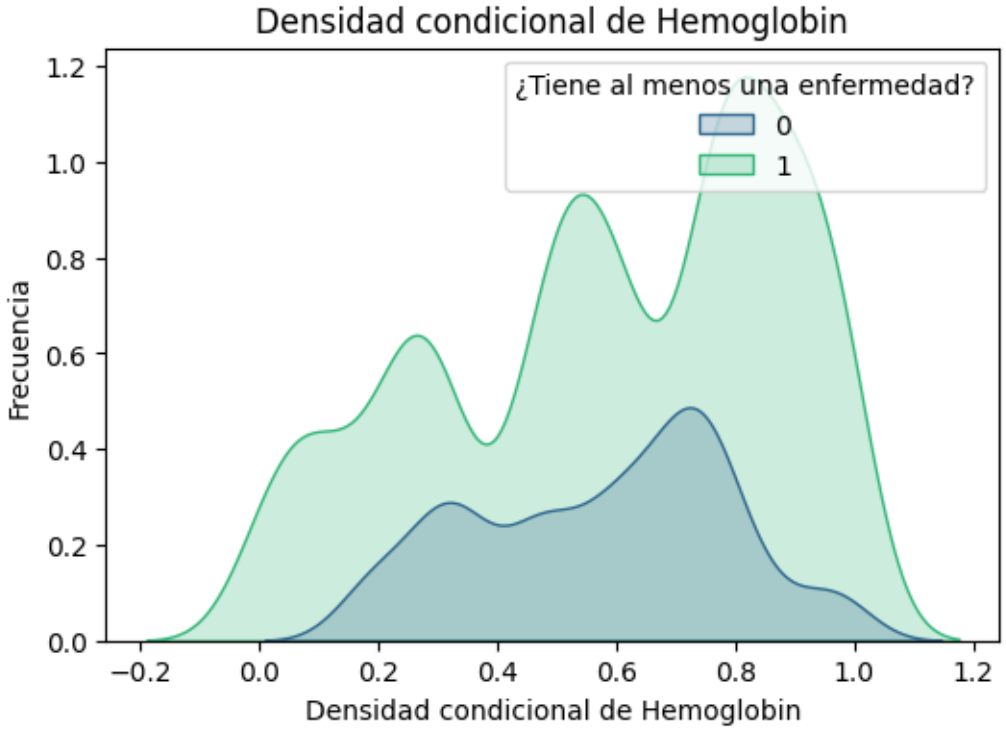
[25]:
```
# Definir el tamaño de la figura
plt.figure(figsize=(6, 6))

# Iterar sobre cada columna del DataFrame
for column in df_new.columns:
    # Crear una nueva figura para cada columna
    plt.figure(figsize=(6, 4))

    # Generar el histplot para la columna actual
    plot = sns.kdeplot(data=df_new, x=column, hue='Disease2', fill=True,␣
 ↪palette='viridis')

    # Cambiar el título del "hue" en la leyenda
    plot.legend_.set_title('¿Tiene al menos una enfermedad?')

    # Agregar título y etiquetas
    plt.title(f'Densidad condicional de {column}')
    plt.xlabel(f'Densidad condicional de {column}')
    plt.ylabel('Frecuencia')

    # Mostrar el gráfico
```

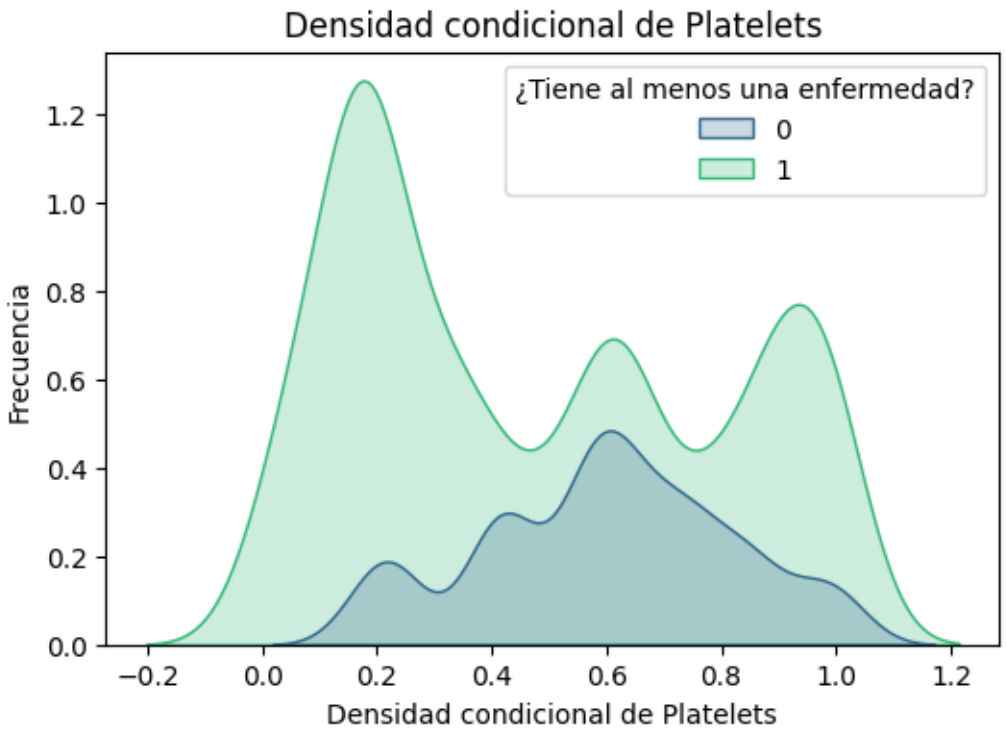
```
    plt.show()
```

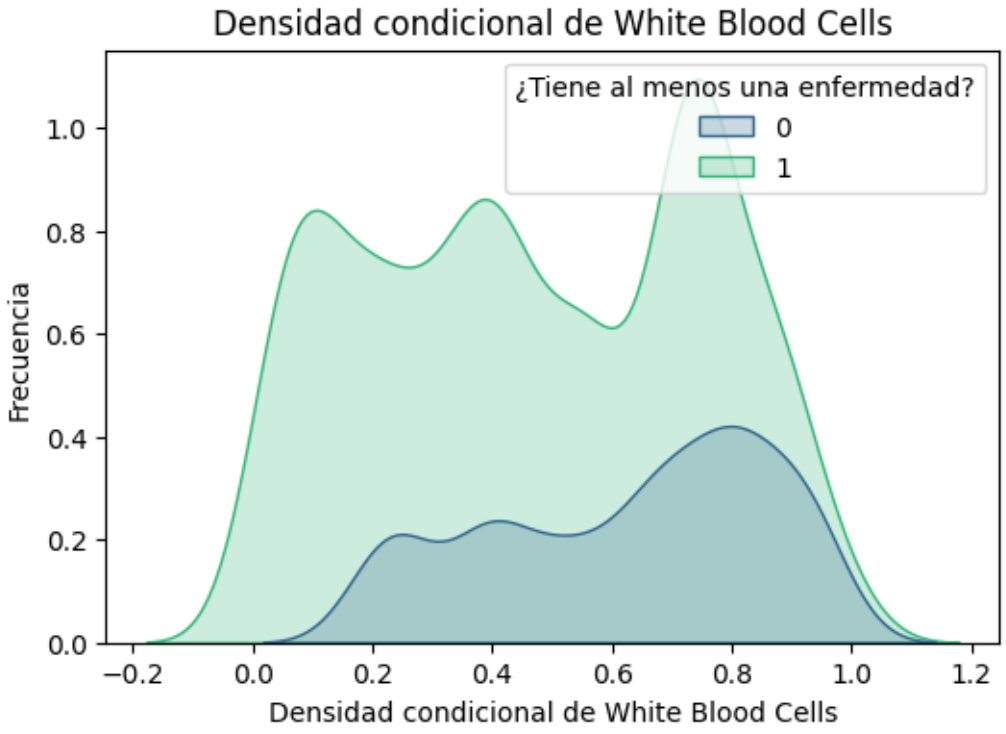
```
<Figure size 600x600 with 0 Axes>
```

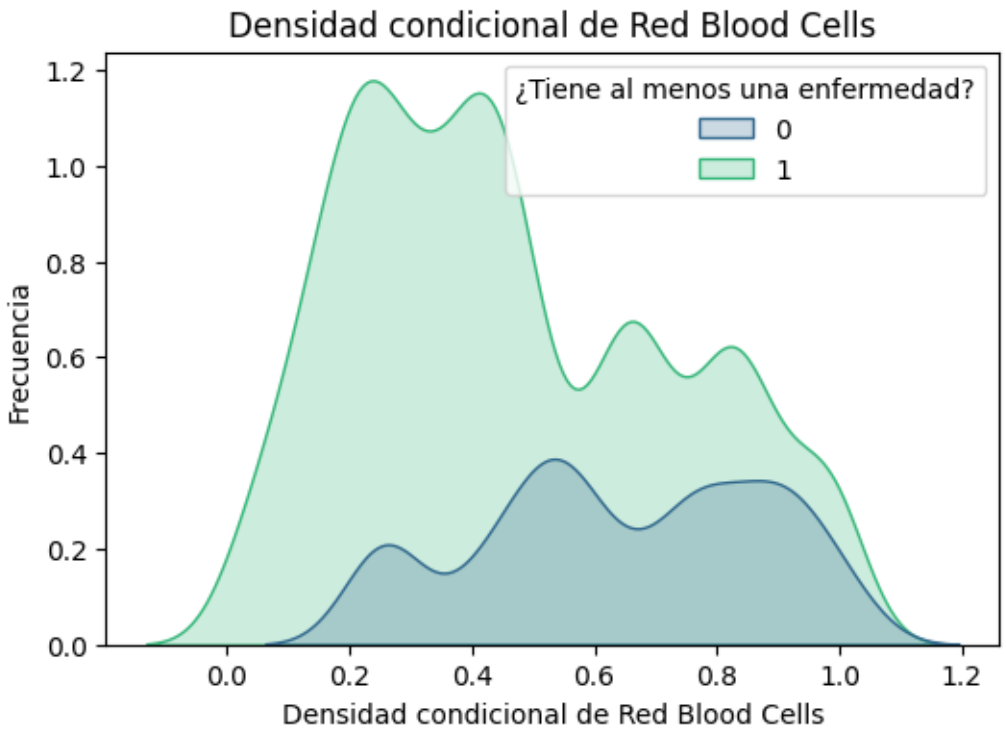


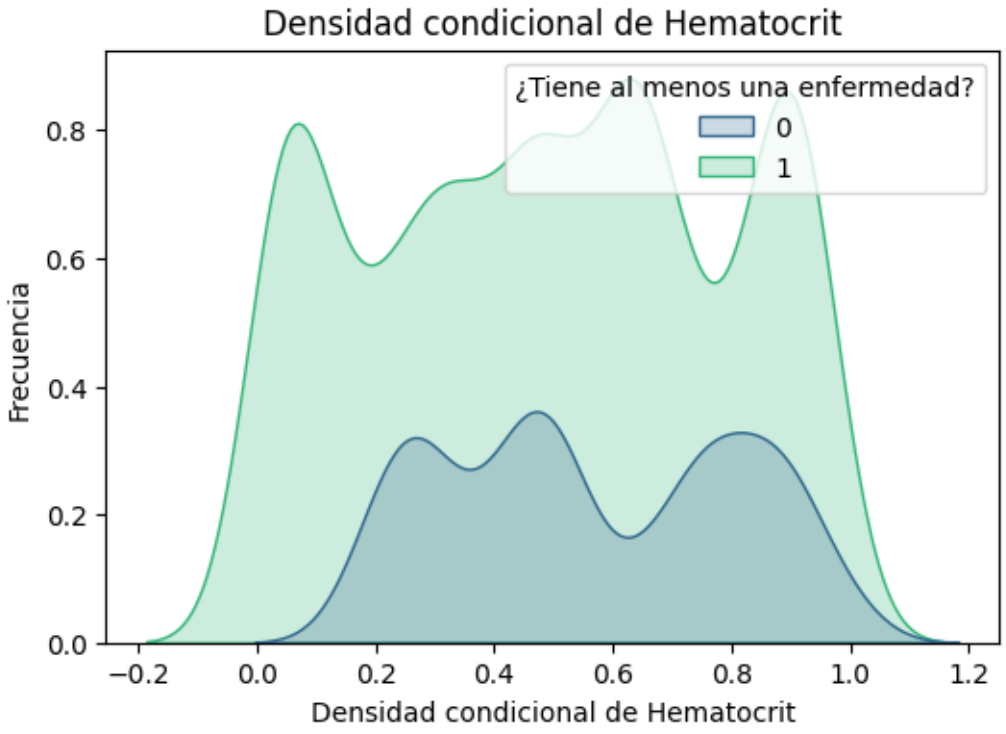

Densidad condicional de Cholesterol
¿Tiene al menos una enfermedad?
0
1
Frecuencia
Densidad condicional de Cholesterol

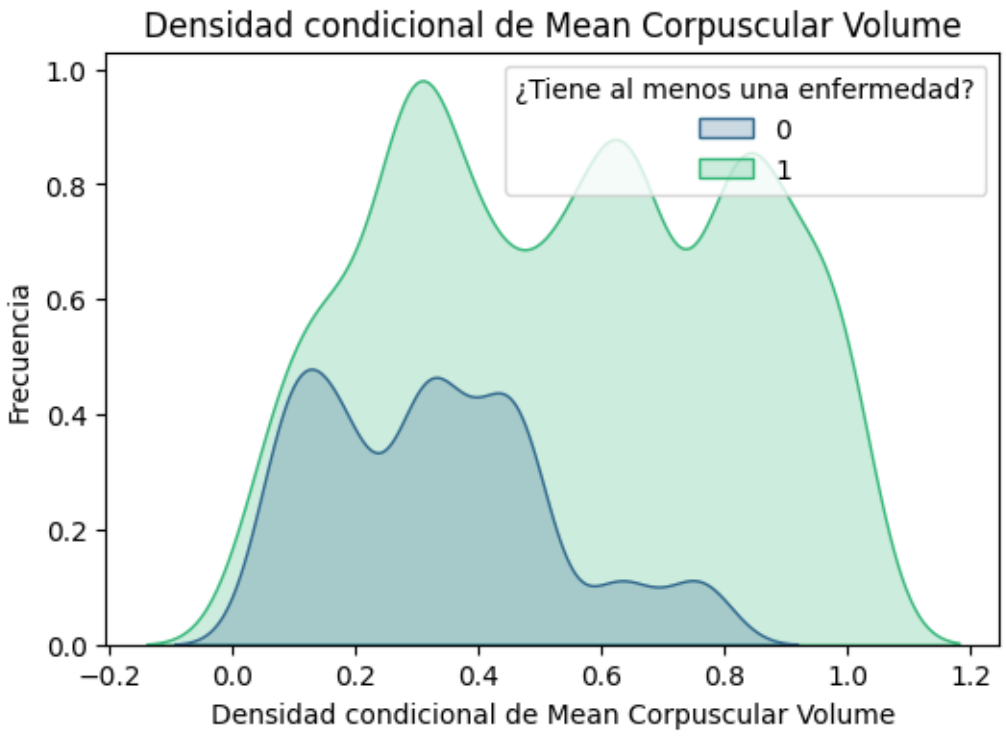

Densidad condicional de Hemoglobin
¿Tiene al menos una enfermedad?
0
1
Frecuencia
Densidad condicional de Hemoglobin

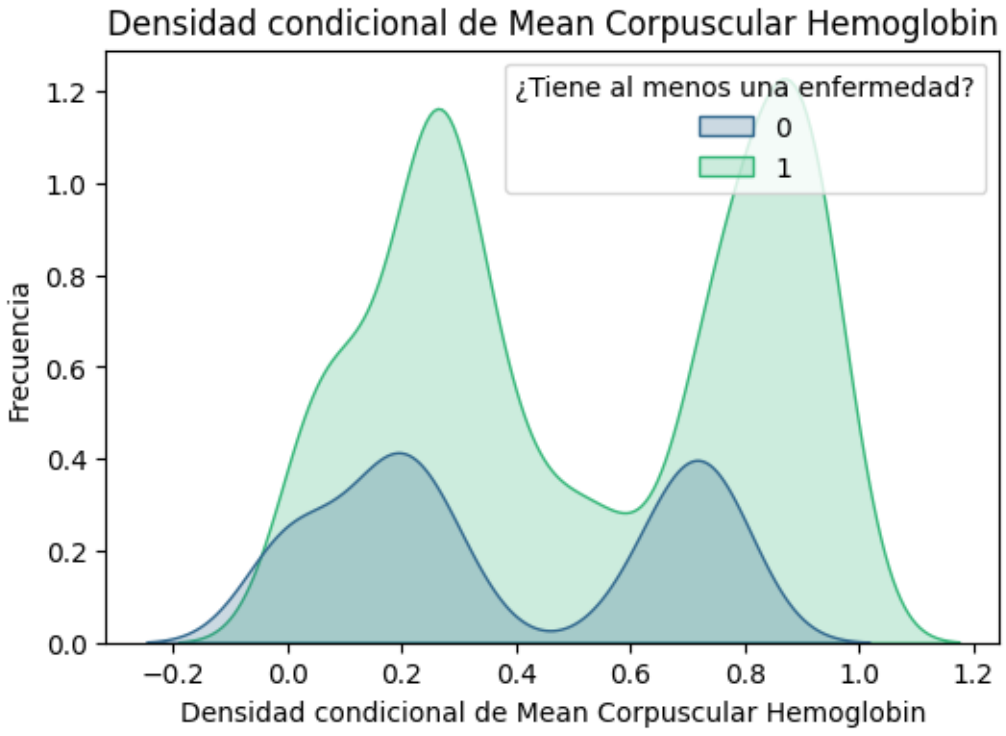

Densidad condicional de Platelets
¿Tiene al menos una enfermedad?
0
1
Frecuencia
0.0
0.2
0.4
0.6
0.8
1.0
1.2
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Densidad condicional de Platelets

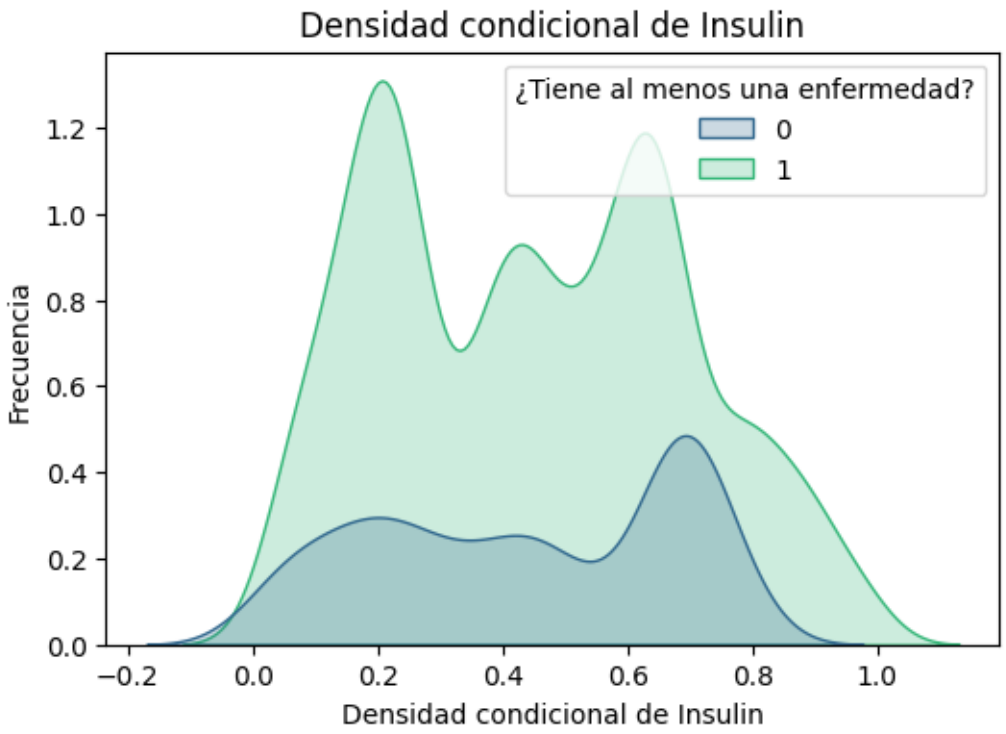

Densidad condicional de White Blood Cells
¿Tiene al menos una enfermedad?
0
1
Frecuencia
0.0
0.2
0.4
0.6
0.8
1.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Densidad condicional de White Blood Cells

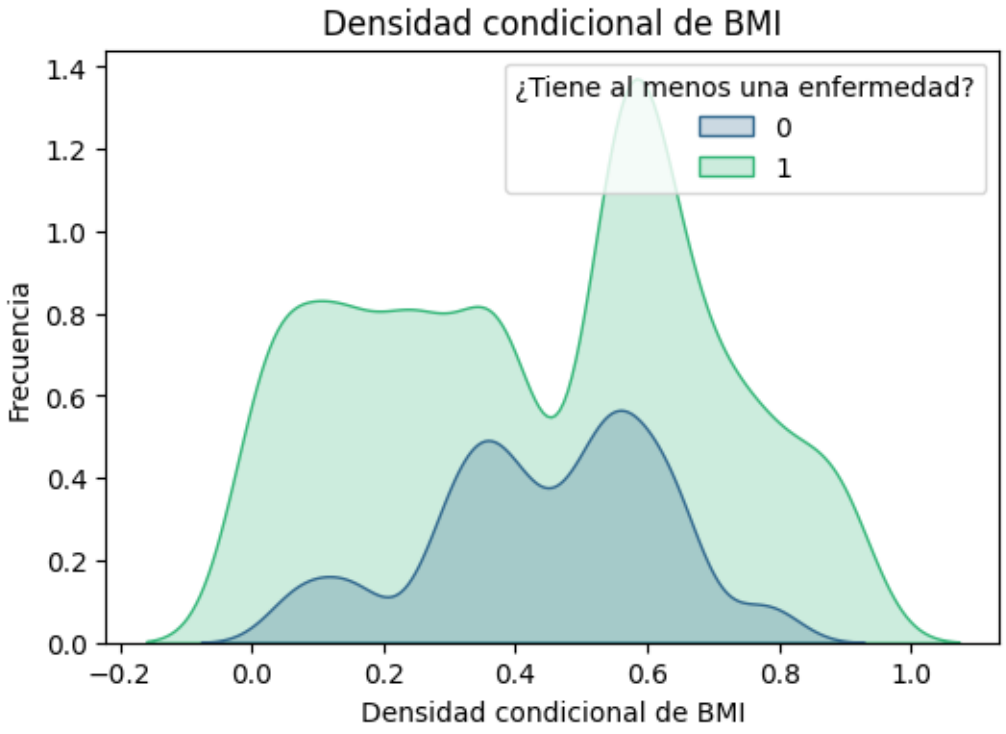

Densidad condicional de Red Blood Cells
¿Tiene al menos una enfermedad?
0
1
Frecuencia
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Densidad condicional de Red Blood Cells

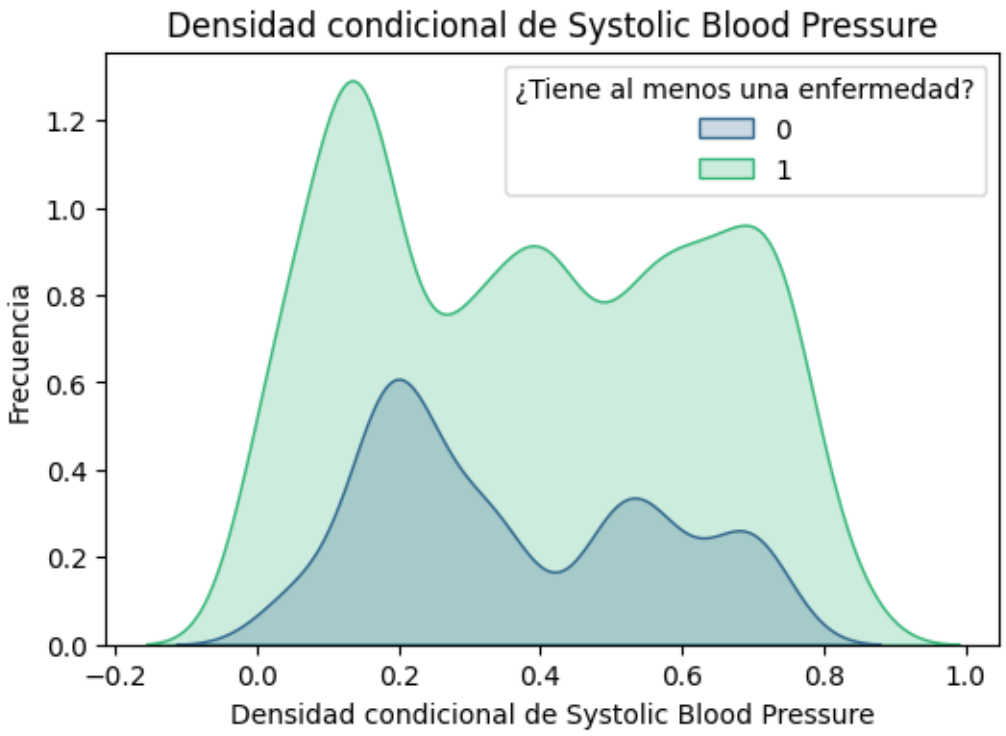

Densidad condicional de Hematocrit
¿Tiene al menos una enfermedad?
0
1
Frecuencia
0.0
0.2
0.4
0.6
0.8
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Densidad condicional de Hematocrit

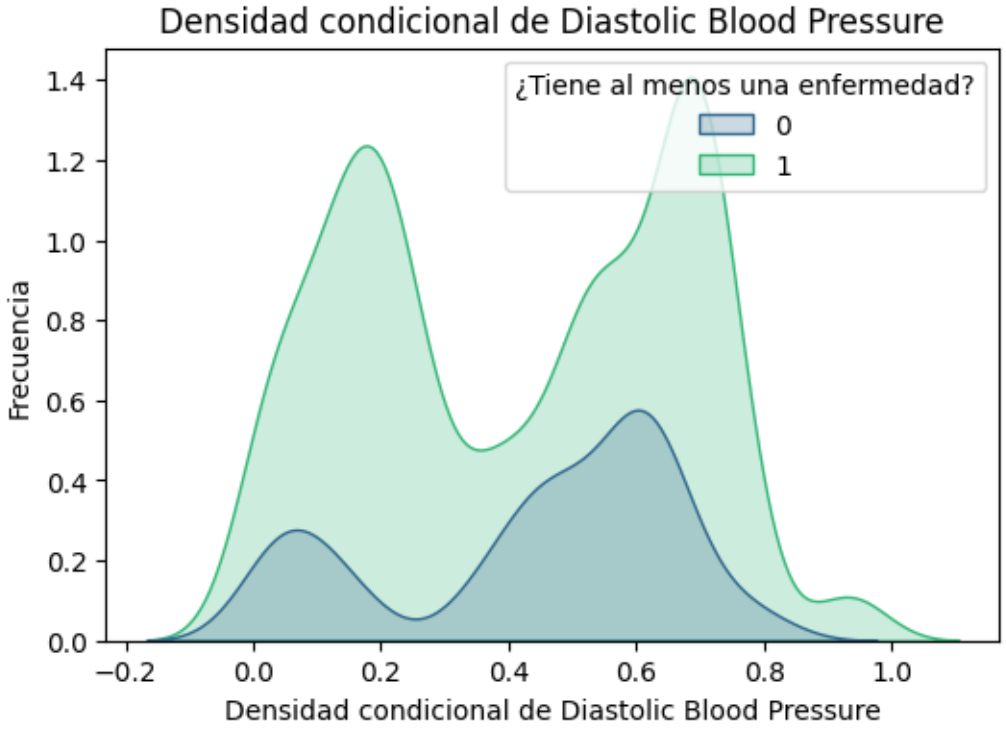

Densidad condicional de Mean Corpuscular Volume
¿Tiene al menos una enfermedad?
0
1
Frecuencia
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Densidad condicional de Mean Corpuscular Volume

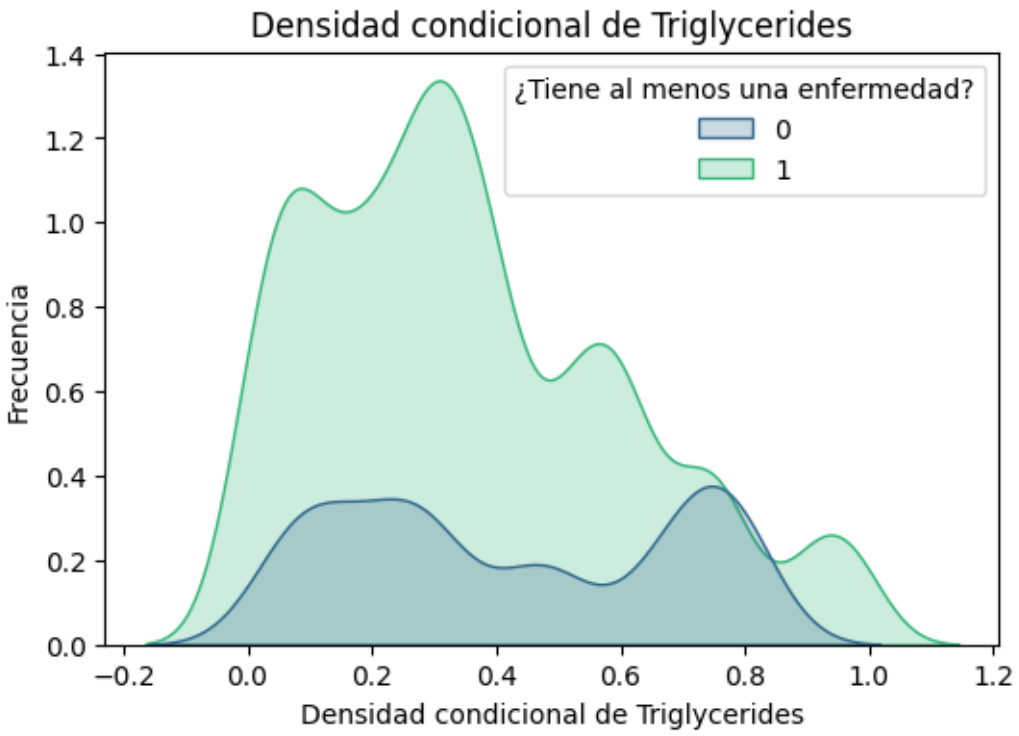

Densidad condicional de Mean Corpuscular Hemoglobin
¿Tiene al menos una enfermedad?
0
1
Frecuencia
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Densidad condicional de Mean Corpuscular Hemoglobin

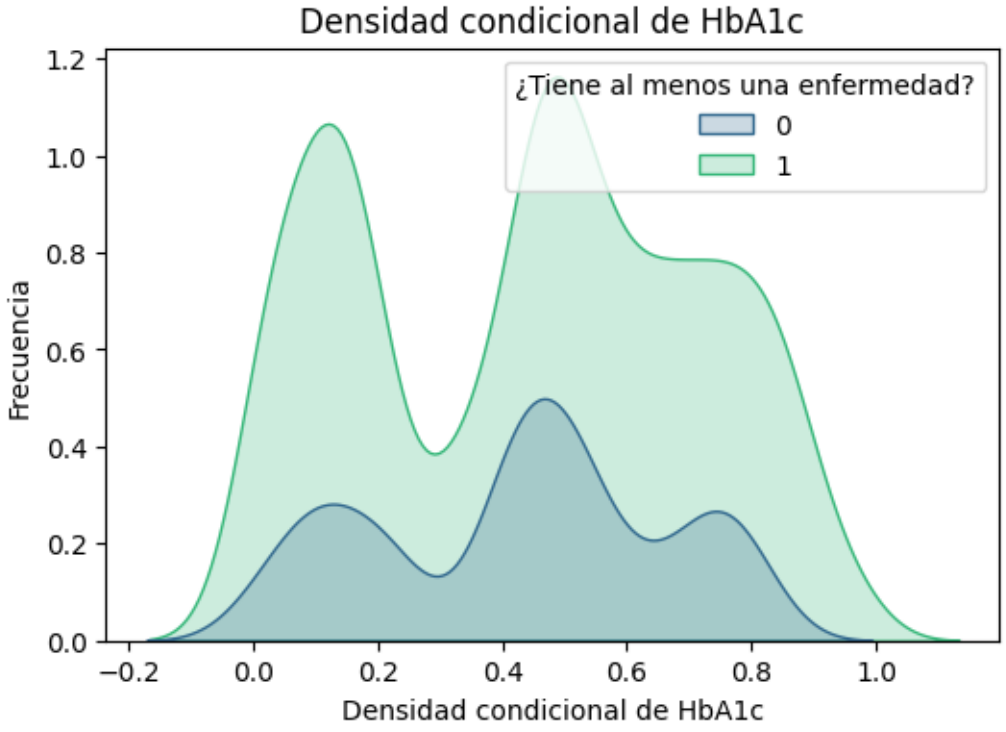

Densidad condicional de Insulin
¿Tiene al menos una enfermedad?
0
1
Frecuencia
0.0
0.2
0.4
0.6
0.8
1.0
1.2
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Densidad condicional de Insulin

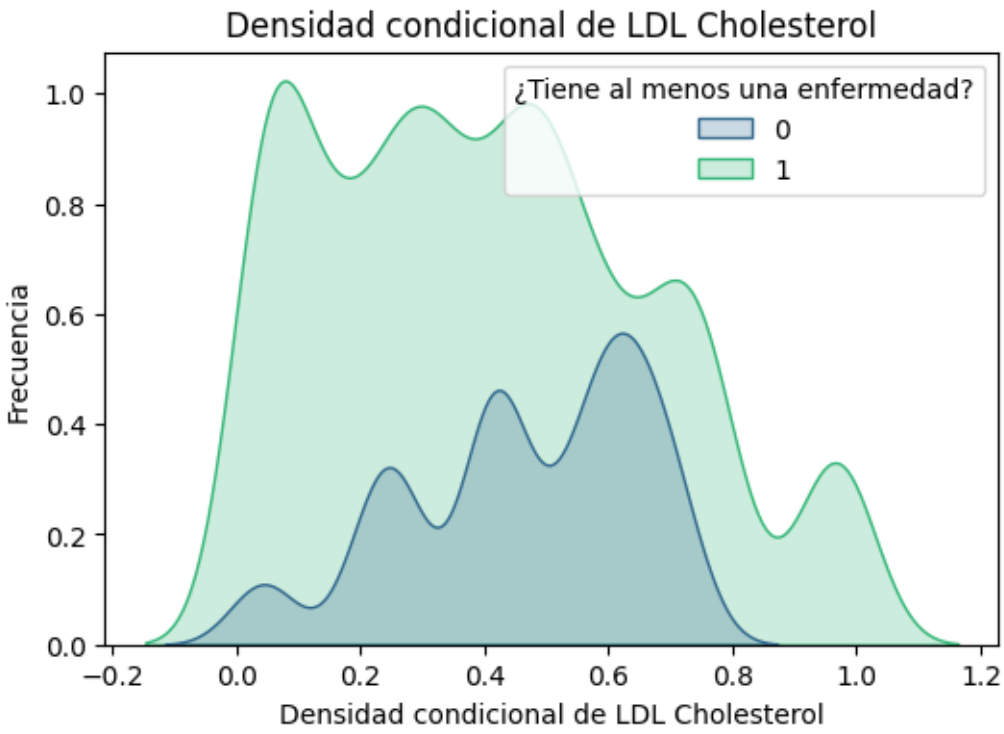

Densidad condicional de BMI
¿Tiene al menos una enfermedad?
0
1
Frecuencia
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Densidad condicional de BMI

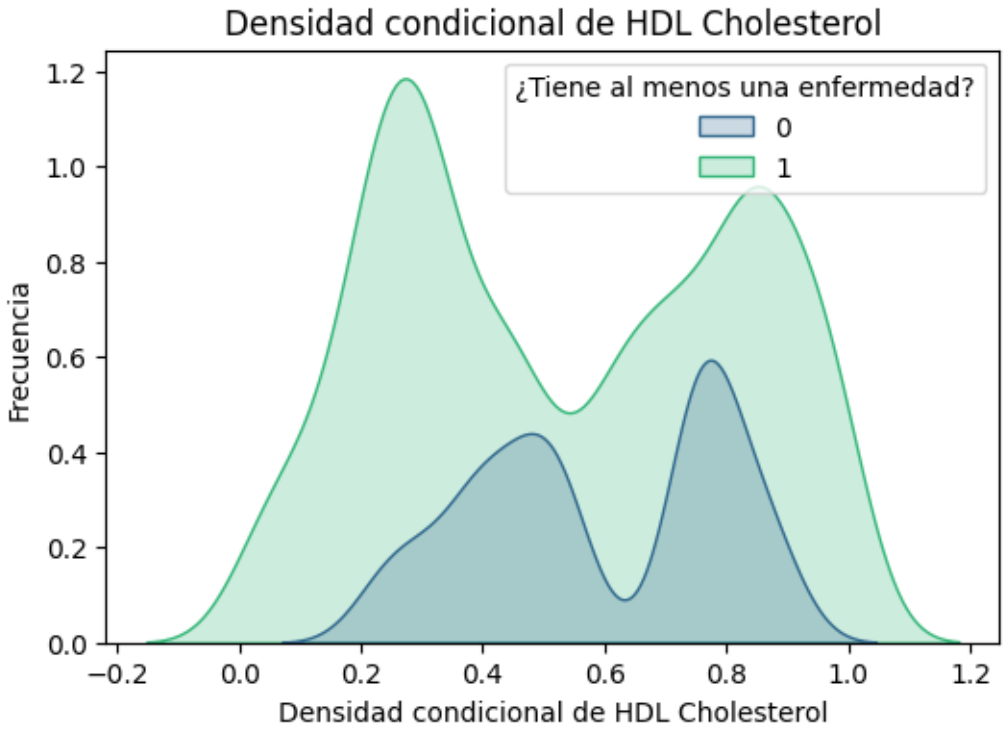

Densidad condicional de Systolic Blood Pressure
¿Tiene al menos una enfermedad?
0
1
Frecuencia
0.0
0.2
0.4
0.6
0.8
1.0
1.2
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Densidad condicional de Systolic Blood Pressure

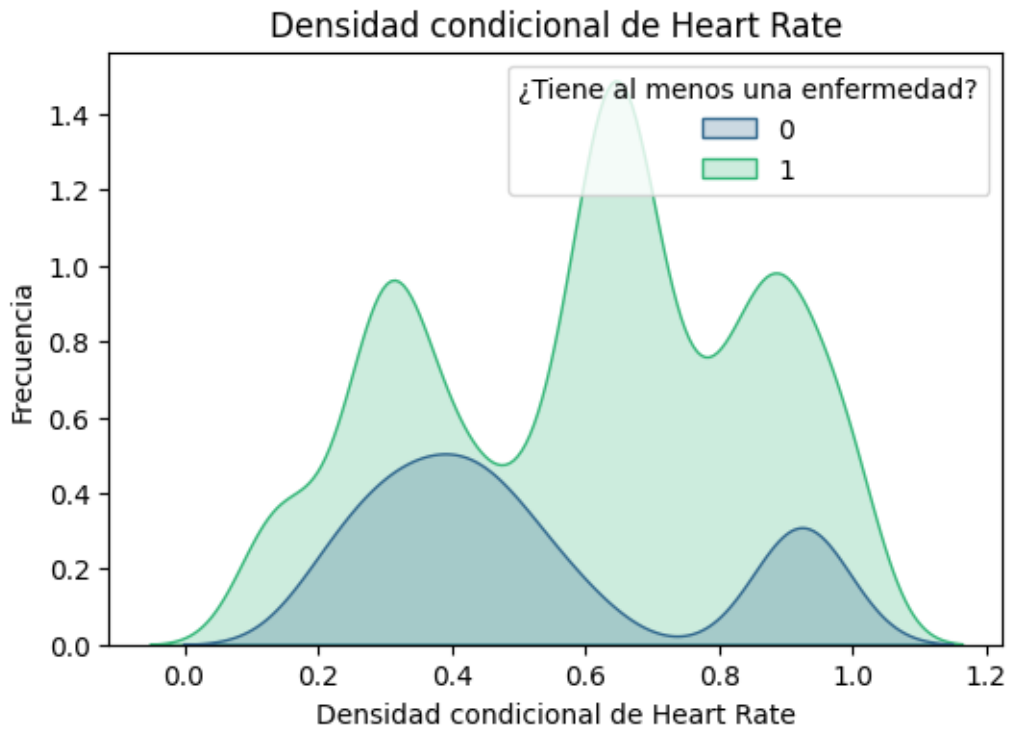

Densidad condicional de Diastolic Blood Pressure
¿Tiene al menos una enfermedad?
0
1
Frecuencia
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Densidad condicional de Diastolic Blood Pressure

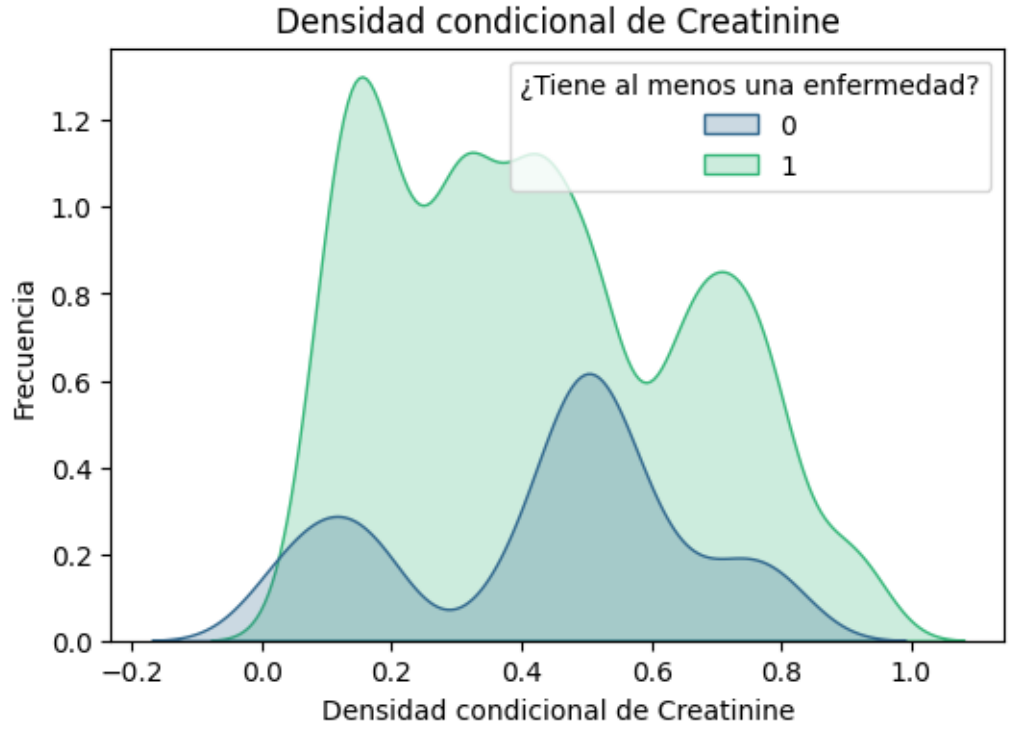

Densidad condicional de Triglycerides
¿Tiene al menos una enfermedad?
0
1
Frecuencia
Densidad condicional de Triglycerides
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2

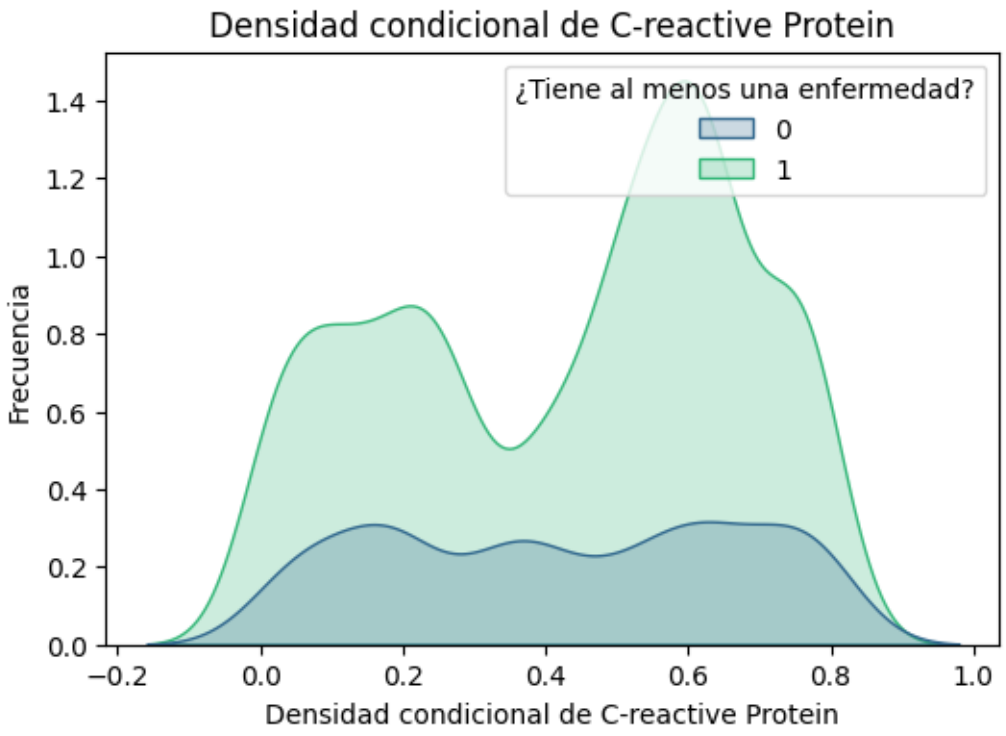

Densidad condicional de HbA1c
¿Tiene al menos una enfermedad?
0
1
Frecuencia
Densidad condicional de HbA1c
0.0
0.2
0.4
0.6
0.8
1.0
1.2
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


Densidad condicional de LDL Cholesterol
¿Tiene al menos una enfermedad?
0
1
Frecuencia
Densidad condicional de LDL Cholesterol

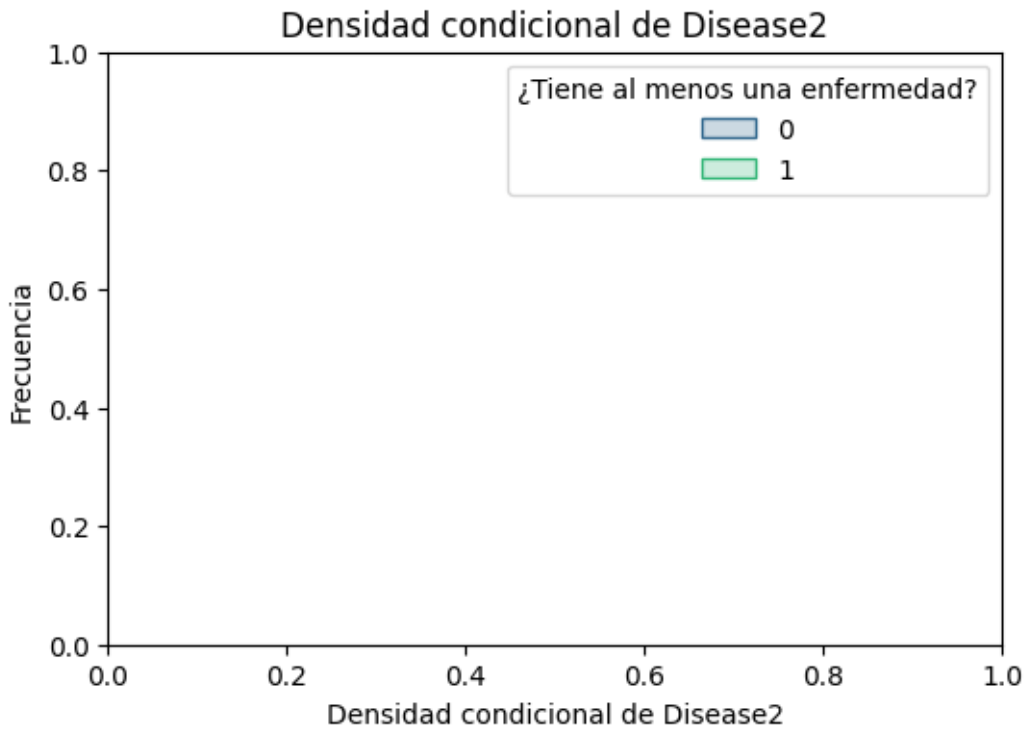

Densidad condicional de HDL Cholesterol
¿Tiene al menos una enfermedad?
0
1
Frecuencia
Densidad condicional de HDL Cholesterol


Densidad condicional de Heart Rate
¿Tiene al menos una enfermedad?
0
1
Frecuencia
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Densidad condicional de Heart Rate


Densidad condicional de Creatinine
¿Tiene al menos una enfermedad?
0
1
Frecuencia
0.0
0.2
0.4
0.6
0.8
1.0
1.2
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Densidad condicional de Creatinine

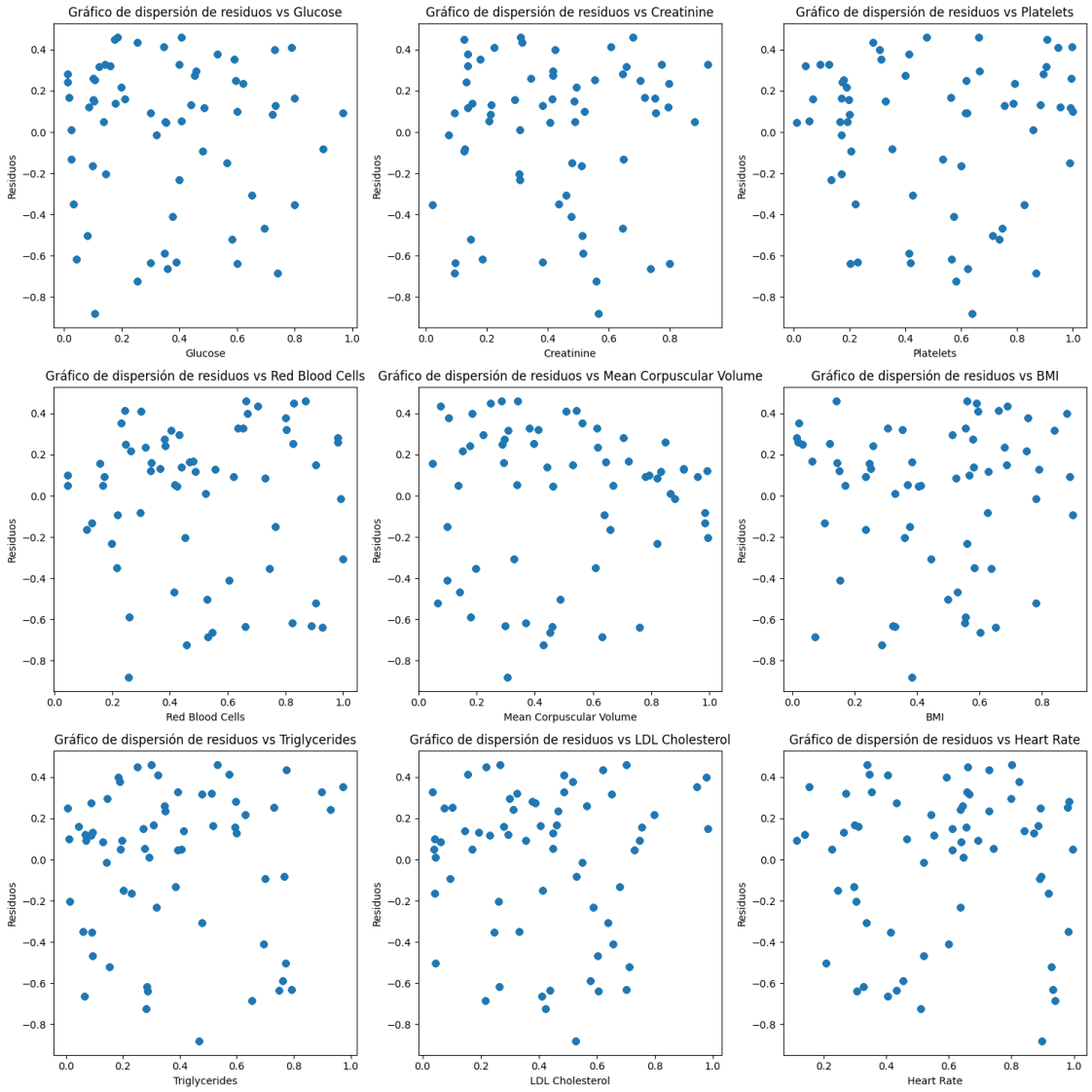



```
/var/folders/7h/czwz4td93vqcp8z2gbys6vfr0000gn/T/ipykernel_26488/2980476353.py:1
0: UserWarning: Dataset has 0 variance; skipping density estimate. Pass
`warn_singular=False` to disable this warning.
  plot = sns.kdeplot(data=df_new, x=column, hue='Disease2', fill=True,
palette='viridis')
```



Criterio de heterocedasticidad: Ninguna variable aporta heterocedasticidad al modelo

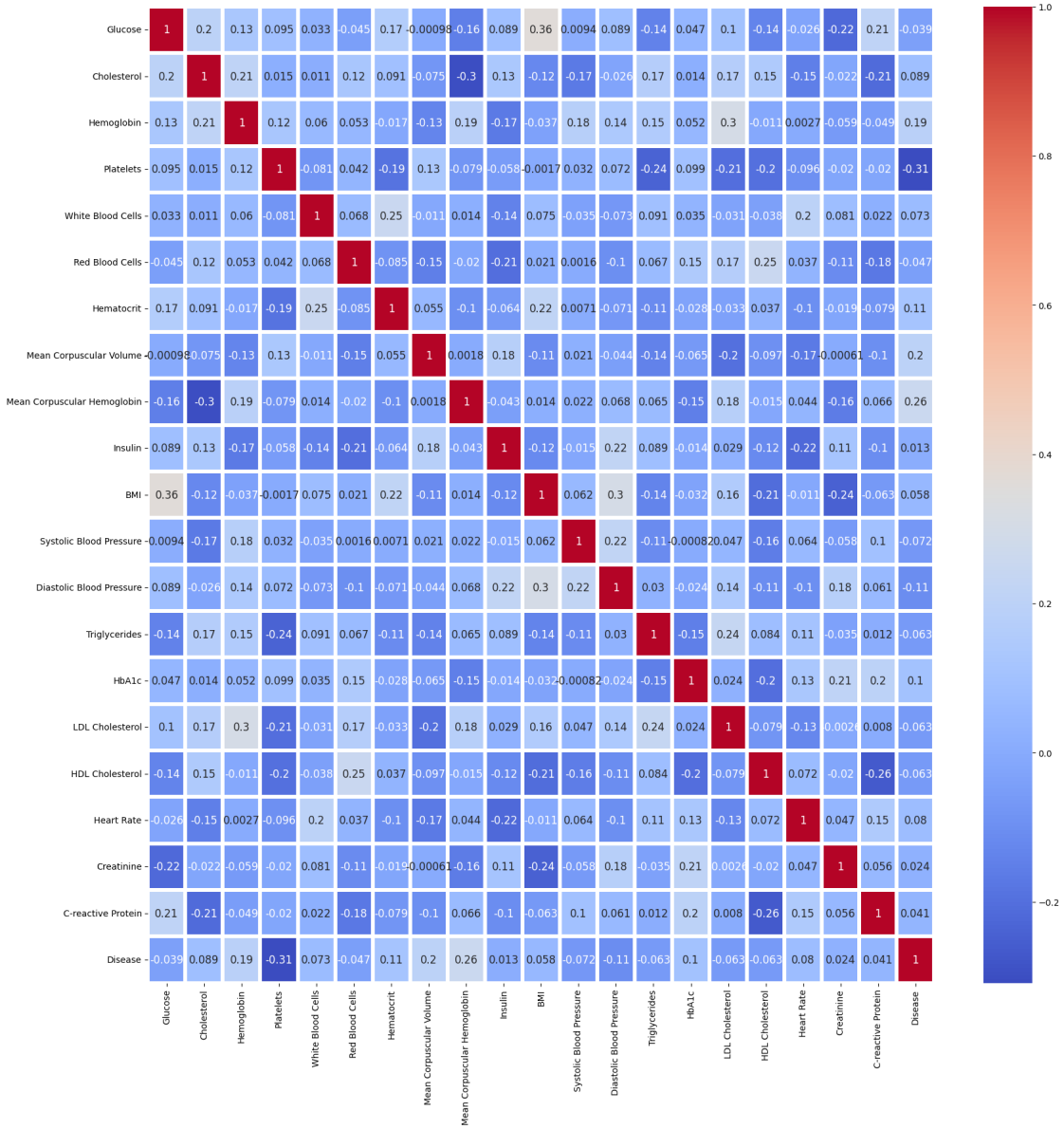
[26]:
```
#Criterio de heterocedasticidad: Ninguna variable aporta heterocedasticidad al␣
 ↪modelo

# Configura los subplots
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 15))

# Itera sobre las columnas de X y los subplots correspondientes
index = 0
for i in range(1, len(X.columns)):  # Excluyendo la constante
    row = index // 3
    col = index % 3
    axes[row, col].scatter(X.iloc[:, i], residuals)  # Excluye la constante
    axes[row, col].set_xlabel(X.columns[i])  # Excluye la constante
    axes[row, col].set_ylabel('Residuos')
    axes[row, col].set_title('Gráfico de dispersión de residuos vs ' + X.
 ↪columns[i])
    index += 1

plt.tight_layout()  # Ajusta automáticamente los subplots para evitar␣
 ↪superposiciones
plt.show()
```



Criterio de matriz de correlacion para seleccion de variables en modelo poisson y binomial negativa

[27]:
```
plt.figure(figsize=(20,20))
sns.heatmap(df_2.corr(), cmap='coolwarm', annot=True, linecolor='white',␣
 ↪linewidths=4, annot_kws={"fontsize":12})
```

[27]: <Axes: >



Criterio de distribuciones de los datos en comparacion a la variable dependiente

[28]:
```
var_elejidas2 = ['Glucose', 'Cholesterol', 'Hemoglobin', 'Platelets', 'White␣
↪Blood Cells', 'Red Blood Cells',
                 'Hematocrit', 'Mean Corpuscular Volume', 'Mean Corpuscular␣
↪Hemoglobin', 'Insulin', 'BMI',
                 'Systolic Blood Pressure', 'Diastolic Blood Pressure',␣
↪'Triglycerides', 'HbA1c', 'LDL Cholesterol',
                 'HDL Cholesterol', 'Heart Rate', 'Creatinine', 'C-reactive␣
↪Protein']
```

```
# Seleccionar las columnas del DataFrame y la variable "Disease"
df_selected = df_2[var_elejidas2 + ['Disease']]

# Apilar las columnas seleccionadas para facilitar la visualización
df_stacked = df_selected.melt(id_vars='Disease', var_name='Variable',␣
 ↪value_name='Valor')

# Crear el gráfico de barras apiladas
plt.figure(figsize=(12, 8))
sns.barplot(data=df_stacked, x='Variable', y='Valor', hue='Disease',␣
 ↪palette='bright')
plt.xticks(rotation=45, ha='right')
plt.xlabel('Variable')
plt.ylabel('Valor')
plt.title('Comparación de Variables por Enfermedad')
plt.legend(title='Disease', loc='upper right')
plt.tight_layout()
plt.show()
```