

```
In [1]: from io import BytesIO
        from zipfile import ZipFile
        import pandas as pd
        import numpy as np
        import os
        import datetime
```

```
In [2]: path = "/home/nacho/Documents/coronavirus/COVID-19_Paper/"
        os.chdir(os.path.join(path))
```

```
In [3]: df = pd.read_csv("datos_abiertos_covid19.zip")
```

```
In [13]: list(df)
```

```
Out[13]: ['FECHA_ACTUALIZACION',  
          'ID_REGISTRO',  
          'ORIGEN',  
          'SECTOR',  
          'ENTIDAD_UM',  
          'SEXO',  
          'ENTIDAD_NAC',  
          'ENTIDAD_RES',  
          'MUNICIPIO_RES',  
          'TIPO_PACIENTE',  
          'FECHA_INGRESO',  
          'FECHA_SINTOMAS',  
          'FECHA_DEF',  
          'INTUBADO',  
          'NEUMONIA',  
          'EDAD',  
          'NACIONALIDAD',  
          'EMBARAZO',  
          'HABLA_LENGUA_INDIG',  
          'INDIGENA',  
          'DIABETES',  
          'EPOC',  
          'ASMA',  
          'INMUSUPR',  
          'HIPERTENSION',  
          'OTRA_COM',  
          'CARDIOVASCULAR',  
          'OBESIDAD',  
          'RENAL_CRONICA',  
          'TABAQUISMO',  
          'OTRO_CASO',  
          'TOMA_MUESTRA_LAB',  
          'RESULTADO_LAB',  
          'TOMA_MUESTRA_ANTIGENO',  
          'RESULTADO_ANTIGENO',  
          'CLASIFICACION_FINAL',  
          'MIGRANTE',  
          'PAIS_NACIONALIDAD',  
          'PAIS_ORIGEN',  
          'UCI']
```

```

In [4]: def filter_exclude_columns(df):
        df.drop(['FECHA_ACTUALIZACION', 'ID_REGISTRO', 'ORIGEN', 'MIGRANTE', 'PAIS_ORIGEN', 'PAIS_NACIONALIDAD', 'MUNICIPIO_RES', 'ENTIDAD_NAC', 'NACIONALIDAD', 'HABLA_LENGUA_INDIG', 'INDIGENA', 'TOMA_MUESTRA_LAB', 'RESULTADO_LAB', 'TOMA_MUESTRA_ANTIGENO', 'RESULTADO_ANTIGENO'], axis=1, inplace = True) #Se eliminan las columnas innecesarias
        return df

def date_preprocessing(df):
    #convierte a tipo fecha fecha_sintoma
    df['FECHA_SINTOMAS'] = pd.to_datetime(df['FECHA_SINTOMAS'])
    #restar columna FECHA_INGRESO menos FECHA_SINTOMAS y guardar en columna dias_dif
    df['FECHA_INGRESO'] = pd.to_datetime(df['FECHA_INGRESO'])
    df['DIAS_DIF_HOSP'] = (df['FECHA_INGRESO'] - df['FECHA_SINTOMAS'])
    df.DIAS_DIF_HOSP = df.DIAS_DIF_HOSP.dt.days
    #eliminar todos los dias negativos
    df.drop(df[df['DIAS_DIF_HOSP'] < 0].index, inplace = True)
    #verificacion
    df['DIAS_DIF_HOSP'][df['DIAS_DIF_HOSP'] < 0]
    #df['DIAS_DIF_HOSP'].astype(int)
    return df

def filter_negative_dates(df):
    #hace una copia ed fecha_def a dias_dif_def
    df['BOOL_DEF'] = df["FECHA_DEF"].copy()
    #CREAR COLUMNA DE NUMERO DE DIAS DE SINTOMAS A FALLECIMIENTO
    #crea columna dias desde sintomas a fallecido
    #reemplazar en fecha_def 9999 con nan
    df["FECHA_DEF"] = df['FECHA_DEF'].replace(['9999-99-99'], np.nan)
    #convertir fecha_def a tipo de dato fecha
    df['FECHA_DEF'] = pd.to_datetime(df['FECHA_DEF'])
    #restar defcha def menos fecha_sintomas
    df['DIAS_DIF_DEF'] = (df['FECHA_DEF'] - df['FECHA_SINTOMAS'])
    df.DIAS_DIF_DEF = df.DIAS_DIF_DEF.dt.days
    df['DIAS_DIF_DEF'] = df['DIAS_DIF_DEF'].replace([np.nan], 0)
    df['DIAS_DIF_DEF'] = df['DIAS_DIF_DEF'].astype(int)
    #validar si hay dias negativos y eliminarlos
    df.drop(df[df['DIAS_DIF_DEF'] < 0].index, inplace = True)
    #verificacion
    df['DIAS_DIF_DEF'][df['DIAS_DIF_DEF'] < 0]
    return df

def filter_binary_status(df):
    #columna defuncion cambia a 0 a los no fallecidos
    df['BOOL_DEF'] = df['BOOL_DEF'].replace(['9999-99-99'], 0)
    #columna defuncion cambia a 1 los fallecidos
    df.loc[df["BOOL_DEF"] != 0, ["BOOL_DEF"]] = 1
    return df

def confirmed_covid(df):
    df['RESULTADO'] = df['CLASIFICACION_FINAL'].copy()
    df.drop(df['CLASIFICACION_FINAL'])
    dictionary = ['RESULTADO']

```

```

for condition in dictionary:
    df.loc[df[condition] != 3, [condition]] = 0
df.loc[df[condition] == 3, [condition]] = 1
return df

def preprocessing(df_prediction):
    df_prediction = df_prediction[df_prediction['RESULTADO'] == 1] #filtrar solo gente positiva covid
    df_prediction.loc[df_prediction['SEXO'] == 2, ['SEXO']] = 0 #Hombre es 1, Mujer es 0
    df_prediction.loc[df_prediction['EMBARAZO'] == 97, ['EMBARAZO']] = 2
    df_prediction.loc[df_prediction['EMBARAZO'] == 98, ['EMBARAZO']] = 2
    df_prediction.loc[df_prediction['EMBARAZO'] == 99, ['EMBARAZO']] = 2
    #eliminar los hombres embarazados
    df_prediction.drop(df_prediction[(df_prediction['SEXO'] == 0) & (df_prediction['EMBARAZO'] == 1)].index, inplace = True)
    #filtra tipo paciente en 0:No hosp, 1:Si hosp
    df_prediction.loc[df_prediction['TIPO_PACIENTE'] == 1, ['TIPO_PACIENTE']] = 0
    df_prediction.loc[df_prediction['TIPO_PACIENTE'] == 2, ['TIPO_PACIENTE']] = 1
    return df_prediction

```

```

In [5]: def print_values(conditions, df):
        for i in df[conditions]:
            #x = df[i].value_counts()
            x = (df[i].value_counts()/df[i].count())*100
            print(i, "\n", x)
            print()

```

```

In [6]: #preprocessing
df = filter_exclude_columns(df)
df = date_preprocessing(df)
df = filter_negative_dates(df)
df = filter_binary_status(df)
df = confirmed_covid(df)
df = preprocessing(df)

```

```
In [19]: list(df)
```

```
Out[19]: ['SECTOR',  
          'ENTIDAD_UM',  
          'SEXO',  
          'ENTIDAD_RES',  
          'TIPO_PACIENTE',  
          'FECHA_INGRESO',  
          'FECHA_SINTOMAS',  
          'FECHA_DEF',  
          'INTUBADO',  
          'NEUMONIA',  
          'EDAD',  
          'EMBARAZO',  
          'DIABETES',  
          'EPOC',  
          'ASMA',  
          'INMUSUPR',  
          'HIPERTENSION',  
          'OTRA_COM',  
          'CARDIOVASCULAR',  
          'OBESIDAD',  
          'RENAL_CRONICA',  
          'TABAQUISMO',  
          'OTRO_CASO',  
          'CLASIFICACION_FINAL',  
          'UCI',  
          'DIAS_DIF_HOSP',  
          'BOOL_DEF',  
          'DIAS_DIF_DEF',  
          'RESULTADO']
```

```
In [21]: print("Numero de casos positivos de COVID: ", len(df))  
print("Numero de hospitalizados por COVID: ", df.TIPO_PACIENTE.value_  
counts()[1])  
print("Numero de intubados por COVID: ", df.INTUBADO.value_counts()[1])  
print("Numero de fallecidos por COVID: ", df.BOOL_DEF.value_counts()[1])  
print("Numero de UCI por COVID: ", df.UCI.value_counts()[1])
```

```
Numero de casos positivos de COVID: 2039782  
Numero de hospitalizados por COVID: 386122  
Numero de intubados por COVID: 50199  
Numero de fallecidos por COVID: 183922  
Numero de UCI por COVID: 30282
```

```
In [38]: #CASO 1: si el paciente contagiado de CoV-2 necesitará hospitalización
```

```
In [81]: df_casol = df.copy()
df_casol.drop(df_casol[(df_casol['TIPO_PACIENTE'] == 99)].index, inplace = True)

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA']
for condition in conditions:
    df_casol = df_casol.loc[~((df_casol[condition] == 97) | (df_casol[condition] == 98) | (df_casol[condition] == 99))]
    df_casol.loc[df_casol[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

final_casol_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'TIPO_PACIENTE']
df_casol = df_casol[df_casol.columns.intersection(final_casol_columns)]

#print_values(final_casol_columns, df_casol)
print_values(['TIPO_PACIENTE'], df_casol)
print(df_casol.shape)
```

```
TIPO_PACIENTE
0      81.110739
1      18.889261
Name: TIPO_PACIENTE, dtype: float64

(2031890, 13)
```

```
In [35]: #CASO 2: predecir en base a los descriptores la mortalidad antes del
          hospital (con filtro)
```

```
In [82]: df_caso2 = df.copy()
df_caso2.drop(df_caso2[(df_caso2['TIPO_PACIENTE'] == 99)].index, inplace = True)
#elimina TIPO_PACIENTE = 1 (gente hospitalizada)
df_caso2.drop(df_caso2[(df_caso2['TIPO_PACIENTE'] == 1)].index, inplace = True) #revisar si mejora el rendimiento

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA']
for condition in conditions:
    df_caso2 = df_caso2.loc[~((df_caso2[condition] == 97) | (df_caso2[condition] == 98) | (df_caso2[condition] == 99))]
    df_caso2.loc[df_caso2[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

final_caso2_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'BOOL_DEF', 'TIPO_PACIENTE']
df_caso2 = df_caso2[df_caso2.columns.intersection(final_caso2_columns)]

#print_values(final_caso2_columns, df_caso2)
print_values(['TIPO_PACIENTE', 'BOOL_DEF'], df_caso2)
print(df_caso2.shape)
```

TIPO_PACIENTE

0 100.0

Name: TIPO_PACIENTE, dtype: float64

BOOL_DEF

0 98.965039

1 1.034961

Name: BOOL_DEF, dtype: float64

(1648081, 14)

```
In [65]: #CASO 2.1: predecir en base a los descriptores la mortalidad (sin filtro)
```

```
In [83]: df_caso2 = df.copy()
df_caso2.drop(df_caso2[(df_caso2['TIPO_PACIENTE'] == 99)].index, inplace = True)

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA']
for condition in conditions:
    df_caso2 = df_caso2.loc[~((df_caso2[condition] == 97) | (df_caso2[condition] == 98) | (df_caso2[condition] == 99))]
    df_caso2.loc[df_caso2[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

final_caso2_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'BOOL_DEF', 'TIPO_PACIENTE']
df_caso2 = df_caso2[df_caso2.columns.intersection(final_caso2_columns)]

#print_values(final_caso2_columns, df_caso2)
print_values(['TIPO_PACIENTE', 'BOOL_DEF'], df_caso2)
print(df_caso2.shape)
```

```
TIPO_PACIENTE
0      81.110739
1      18.889261
Name: TIPO_PACIENTE, dtype: float64
```

```
BOOL_DEF
0      91.01659
1       8.98341
Name: BOOL_DEF, dtype: float64
```

```
(2031890, 14)
```

```
In [ ]: #CASO 3: Mortalidad de los contagiados DESPUES de INTUBADO,UCI (sin filtro)
```



```
In [85]: df_caso3 = df.copy()
df_caso3.drop(df_caso3[(df_caso3['TIPO_PACIENTE'] == 99)].index, inplace = True)

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'INTUBADO', 'UCI']
for condition in conditions:
    df_caso3 = df_caso3.loc[~((df_caso3[condition] == 97) | (df_caso3[condition] == 98) | (df_caso3[condition] == 99))]
    df_caso3.loc[df_caso3[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

final_caso3_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'INTUBADO', 'UCI', 'BOOL_DEF', 'TIPO_PACIENTE']
df_caso3 = df_caso3[df_caso3.columns.intersection(final_caso3_columns)]

#print_values(final_caso3_columns, df_caso3)
print_values(['INTUBADO', 'UCI', 'TIPO_PACIENTE', 'BOOL_DEF'], df_caso3)
print(df_caso3.shape)
```

INTUBADO

0 87.022366

1 12.977634

Name: INTUBADO, dtype: float64

UCI

0 92.203439

1 7.796561

Name: UCI, dtype: float64

TIPO_PACIENTE

1 100.0

Name: TIPO_PACIENTE, dtype: float64

BOOL_DEF

0 57.002508

1 42.997492

Name: BOOL_DEF, dtype: float64

(382720, 16)

```
In [ ]: #CASO 3.1: Mortalidad de los contagiados DESPUES de INTUBADO,UCI (con filtro)
```

```

In [86]: df_caso3 = df.copy()
df_caso3.drop(df_caso3[(df_caso3['TIPO_PACIENTE'] == 99)].index, inplace = True)

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'INTUBADO', 'UCI']
for condition in conditions:
    df_caso3 = df_caso3.loc[~((df_caso3[condition] == 97) | (df_caso3[condition] == 98) | (df_caso3[condition] == 99))]
    df_caso3.loc[df_caso3[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

#elimina no intubado y uci
df_caso3.drop(df_caso3[(df_caso3['INTUBADO'] == 0)].index, inplace = True)
df_caso3.drop(df_caso3[(df_caso3['UCI'] == 0)].index, inplace = True)

final_caso3_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'INTUBADO', 'UCI', 'BOOL_DEF', 'TIPO_PACIENTE']
df_caso3 = df_caso3[df_caso3.columns.intersection(final_caso3_columns)]

#print_values(final_caso3_columns, df_caso3)
print_values(['INTUBADO', 'UCI', 'TIPO_PACIENTE', 'BOOL_DEF'], df_caso3)
print(df_caso3.shape)

```

```

INTUBADO
1    100.0
Name: INTUBADO, dtype: float64

UCI
1    100.0
Name: UCI, dtype: float64

TIPO_PACIENTE
1    100.0
Name: TIPO_PACIENTE, dtype: float64

BOOL_DEF
1    75.649795
0    24.350205
Name: BOOL_DEF, dtype: float64

(15351, 16)

```

```

In [ ]: #CASO 3.2: Mortalidad de los contagiados DESPUES de INTUBADO,UCI (fil
tro solo INTUBADO)

```

```

In [87]: df_caso3 = df.copy()
df_caso3.drop(df_caso3[(df_caso3['TIPO_PACIENTE'] == 99)].index, inplace = True)

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'INTUBADO', 'UCI']
for condition in conditions:
    df_caso3 = df_caso3.loc[~((df_caso3[condition] == 97) | (df_caso3[condition] == 98) | (df_caso3[condition] == 99))]
    df_caso3.loc[df_caso3[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

#elimina no intubado
df_caso3.drop(df_caso3[(df_caso3['INTUBADO'] == 0)].index, inplace = True)

final_caso3_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'INTUBADO', 'UCI', 'BOOL_DEF', 'TIPO_PACIENTE']
df_caso3 = df_caso3[df_caso3.columns.intersection(final_caso3_columns)]

#print_values(final_caso3_columns, df_caso3)
print_values(['INTUBADO', 'UCI', 'TIPO_PACIENTE', 'BOOL_DEF'], df_caso3)
print(df_caso3.shape)

```

```

INTUBADO
1    100.0
Name: INTUBADO, dtype: float64

UCI
0    69.092776
1    30.907224
Name: UCI, dtype: float64

TIPO_PACIENTE
1    100.0
Name: TIPO_PACIENTE, dtype: float64

BOOL_DEF
1    82.395103
0    17.604897
Name: BOOL_DEF, dtype: float64

(49668, 16)

```

```

In [ ]: #CASO 3.3: Mortalidad de los contagiados DESPUES de INTUBADO,UCI (filtro solo UCI)

```

```
In [88]: df_caso3 = df.copy()
df_caso3.drop(df_caso3[(df_caso3['TIPO_PACIENTE'] == 99)].index, inplace = True)

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'INTUBADO', 'UCI']
for condition in conditions:
    df_caso3 = df_caso3.loc[~((df_caso3[condition] == 97) | (df_caso3[condition] == 98) | (df_caso3[condition] == 99))]
    df_caso3.loc[df_caso3[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

#elimina no uci
df_caso3.drop(df_caso3[(df_caso3['UCI'] == 0)].index, inplace = True)

final_caso3_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'INTUBADO', 'UCI', 'BOOL_DEF', 'TIPO_PACIENTE']
df_caso3 = df_caso3[df_caso3.columns.intersection(final_caso3_columns)]

#print_values(final_caso3_columns, df_caso3)
print_values(['INTUBADO', 'UCI', 'TIPO_PACIENTE', 'BOOL_DEF'], df_caso3)
print(df_caso3.shape)
```

```
INTUBADO
1    51.446094
0    48.553906
Name: INTUBADO, dtype: float64

UCI
1    100.0
Name: UCI, dtype: float64

TIPO_PACIENTE
1    100.0
Name: TIPO_PACIENTE, dtype: float64

BOOL_DEF
1    54.61309
0    45.38691
Name: BOOL_DEF, dtype: float64

(29839, 16)
```

```
In [ ]: #CASO 4: Necesidad de UCI ANTES de diagnostico de neumonia (sin filtro)
```

```

In [7]: df_caso4 = df.copy()

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'UCI']
for condition in conditions:
    df_caso4 = df_caso4.loc[~((df_caso4[condition] == 97) | (df_caso4[condition] == 98) | (df_caso4[condition] == 99))]
    df_caso4.loc[df_caso4[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

final_caso4_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'UCI', 'TIPO_PACIENTE']
df_caso4 = df_caso4[df_caso4.columns.intersection(final_caso4_columns)]

#print_values(final_caso4_columns, df_caso4)
print_values(['TIPO_PACIENTE', 'UCI'], df_caso4)
print(df_caso4.shape)

```

```

TIPO_PACIENTE
1      100.0
Name: TIPO_PACIENTE, dtype: float64

```

```

UCI
0      92.203439
1       7.796561
Name: UCI, dtype: float64

```

```

(382720, 14)

```

```

In [ ]: #CASO 5: Necesidad de ICU DESPUES de diagnostico de neumonia (sin filtro)

```

```

In [92]: df_caso5 = df.copy()

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'UCI', 'NEUMONIA']
for condition in conditions:
    df_caso5 = df_caso5.loc[~((df_caso5[condition] == 97) | (df_caso5[condition] == 98) | (df_caso5[condition] == 99))]
    df_caso5.loc[df_caso5[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

final_caso5_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'UCI', 'NEUMONIA', 'TIPO_PACIENTE']
df_caso5 = df_caso5[df_caso5.columns.intersection(final_caso5_columns)]

#print_values(final_caso5_columns, df_caso5)
print_values(['TIPO_PACIENTE', 'NEUMONIA', 'UCI'], df_caso5)
print(df_caso5.shape)

```

TIPO_PACIENTE

1 100.0

Name: TIPO_PACIENTE, dtype: float64

NEUMONIA

1 65.58241

0 34.41759

Name: NEUMONIA, dtype: float64

UCI

0 92.203439

1 7.796561

Name: UCI, dtype: float64

(382720, 15)

```

In [ ]: #CASO 5.1: Necesidad de ICU DESPUES de diagnostico de neumonia (con filtro)

```

```

In [94]: df_caso5 = df.copy()

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'UCI', 'NEUMONIA']
for condition in conditions:
    df_caso5 = df_caso5.loc[~((df_caso5[condition] == 97) | (df_caso5[condition] == 98) | (df_caso5[condition] == 99))]
    df_caso5.loc[df_caso5[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

#elimina no neumonia
df_caso5.drop(df_caso5[(df_caso5['NEUMONIA'] == 0)].index, inplace = True)

final_caso5_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'UCI', 'NEUMONIA', 'TIPO_PACIENTE']
df_caso5 = df_caso5[df_caso5.columns.intersection(final_caso5_columns)]

#print_values(final_caso5_columns, df_caso5)
print_values(['TIPO_PACIENTE', 'NEUMONIA', 'UCI'], df_caso5)
print(df_caso5.shape)

```

```

TIPO_PACIENTE
1      100.0
Name: TIPO_PACIENTE, dtype: float64

```

```

NEUMONIA
1      100.0
Name: NEUMONIA, dtype: float64

```

```

UCI
0      89.766013
1     10.233987
Name: UCI, dtype: float64

```

```

(250997, 15)

```

```

In [ ]: #CASO 6: necesidad de ventilador ANTES de DIAGNOSTICO de neumonia e ICU (sin filtro)

```

```

In [100]: df_caso6 = df.copy()

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'INTUBADO']
for condition in conditions:
    df_caso6 = df_caso6.loc[~((df_caso6[condition] == 97) | (df_caso6[condition] == 98) | (df_caso6[condition] == 99))]
    df_caso6.loc[df_caso6[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

final_caso6_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'INTUBADO', 'TIPO_PACIENTE']
df_caso6 = df_caso6[df_caso6.columns.intersection(final_caso6_columns)]

#print_values(final_caso6_columns, df_caso6)
print_values(['TIPO_PACIENTE', 'INTUBADO'], df_caso6)
print(df_caso6.shape)

```

```

TIPO_PACIENTE
1      100.0
Name: TIPO_PACIENTE, dtype: float64

```

```

INTUBADO
0      87.020547
1      12.979453
Name: INTUBADO, dtype: float64

```

```

(382728, 14)

```

```

In [ ]: #CASO 7: necesidad de ventilador DESPUES de DIAGNOSTICO de neumonia e ICU (sin filtro)

```



```
In [104]: df_caso7 = df.copy()

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'UCI', 'NEUMONIA', 'INTUBADO']
for condition in conditions:
    df_caso7 = df_caso7.loc[~((df_caso7[condition] == 97) | (df_caso7[condition] == 98) | (df_caso7[condition] == 99))]
    df_caso7.loc[df_caso7[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

final_caso7_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'UCI', 'NEUMONIA', 'INTUBADO', 'TIPO_PACIENTE']
df_caso7 = df_caso7[df_caso7.columns.intersection(final_caso7_columns)]

#print_values(final_caso7_columns, df_caso7)
print_values(['TIPO_PACIENTE', 'UCI', 'NEUMONIA', 'INTUBADO'], df_caso7)
print(df_caso7.shape)
```

TIPO_PACIENTE

1 100.0

Name: TIPO_PACIENTE, dtype: float64

UCI

0 92.203439

1 7.796561

Name: UCI, dtype: float64

NEUMONIA

1 65.58241

0 34.41759

Name: NEUMONIA, dtype: float64

INTUBADO

0 87.022366

1 12.977634

Name: INTUBADO, dtype: float64

(382720, 16)

```
In [ ]: #CASO 7.1: necesidad de ventilador DESPUES de DIAGNOSTICO de neumonia e ICU (con filtro)
```

```

In [106]: df_caso7 = df.copy()

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'UCI', 'NEUMONIA', 'INTUBADO']
for condition in conditions:
    df_caso7 = df_caso7.loc[~((df_caso7[condition] == 97) | (df_caso7[condition] == 98) | (df_caso7[condition] == 99))]
    df_caso7.loc[df_caso7[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

#elimina no neumonia e ICU
df_caso7.drop(df_caso7[(df_caso7['NEUMONIA'] == 0)].index, inplace = True)
df_caso7.drop(df_caso7[(df_caso7['UCI'] == 0)].index, inplace = True)

final_caso7_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'UCI', 'NEUMONIA', 'INTUBADO', 'TIPO_PACIENTE']
df_caso7 = df_caso7[df_caso7.columns.intersection(final_caso7_columns)]

#print_values(final_caso7_columns, df_caso7)
print_values(['TIPO_PACIENTE', 'UCI', 'NEUMONIA', 'INTUBADO'], df_caso7)
print(df_caso7.shape)

```

```

TIPO_PACIENTE
1      100.0
Name: TIPO_PACIENTE, dtype: float64

```

```

UCI
1      100.0
Name: UCI, dtype: float64

```

```

NEUMONIA
1      100.0
Name: NEUMONIA, dtype: float64

```

```

INTUBADO
1      56.195741
0      43.804259
Name: INTUBADO, dtype: float64

```

```

(25687, 16)

```

```

In [ ]: #CASO 7.2: necesidad de ventilador DESPUES de DIAGNOSTICO de neumonia e ICU (solo filtro UCI)

```

```

In [108]: df_caso7 = df.copy()

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'UCI', 'NEUMONIA', 'INTUBADO']
for condition in conditions:
    df_caso7 = df_caso7.loc[~((df_caso7[condition] == 97) | (df_caso7[condition] == 98) | (df_caso7[condition] == 99))]
    df_caso7.loc[df_caso7[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

#elimina no ICU
df_caso7.drop(df_caso7[(df_caso7['UCI'] == 0)].index, inplace = True)

final_caso7_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'UCI', 'NEUMONIA', 'INTUBADO', 'TIPO_PACIENTE']
df_caso7 = df_caso7[df_caso7.columns.intersection(final_caso7_columns)]

#print_values(final_caso7_columns, df_caso7)
print_values(['TIPO_PACIENTE', 'UCI', 'NEUMONIA', 'INTUBADO'], df_caso7)
print(df_caso7.shape)

```

```

TIPO_PACIENTE
1      100.0
Name: TIPO_PACIENTE, dtype: float64

```

```

UCI
1      100.0
Name: UCI, dtype: float64

```

```

NEUMONIA
1      86.085325
0      13.914675
Name: NEUMONIA, dtype: float64

```

```

INTUBADO
1      51.446094
0      48.553906
Name: INTUBADO, dtype: float64

```

```

(29839, 16)

```

```

In [ ]: #CASO 7.3: necesidad de ventilador DESPUES de DIAGNOSTICO de neumonia e ICU (solo filtro neumonia)

```

```

In [110]: df_caso7 = df.copy()

conditions = ['EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'UCI', 'NEUMONIA', 'INTUBADO']
for condition in conditions:
    df_caso7 = df_caso7.loc[~((df_caso7[condition] == 97) | (df_caso7[condition] == 98) | (df_caso7[condition] == 99))]
    df_caso7.loc[df_caso7[condition] == 2, [condition]] = 0 #0 es NO, 1 es SI

#elimina no neumonia
df_caso7.drop(df_caso7[(df_caso7['NEUMONIA'] == 0)].index, inplace = True)

final_caso7_columns = ['EDAD', 'EMBARAZO', 'RENAL_CRONICA', 'DIABETES', 'INMUSUPR', 'EPOC', 'OBESIDAD', 'HIPERTENSION', 'TABAQUISMO', 'CARDIOVASCULAR', 'ASMA', 'SEXO', 'UCI', 'NEUMONIA', 'INTUBADO', 'TIPO_PACIENTE']
df_caso7 = df_caso7[df_caso7.columns.intersection(final_caso7_columns)]

#print_values(final_caso7_columns, df_caso7)
print_values(['TIPO_PACIENTE', 'UCI', 'NEUMONIA', 'INTUBADO'], df_caso7)
print(df_caso7.shape)

```

TIPO_PACIENTE

1 100.0

Name: TIPO_PACIENTE, dtype: float64

UCI

0 89.766013

1 10.233987

Name: UCI, dtype: float64

NEUMONIA

1 100.0

Name: NEUMONIA, dtype: float64

INTUBADO

0 83.378287

1 16.621713

Name: INTUBADO, dtype: float64

(250997, 16)

In []: