de-validacion-de-datos-y-variables

November 23, 2023

[49]:
```
import pandas as pd
import os
import re
```

[50]:
```
# ruta de archivos
files = os.listdir("c:\\archivos\proyecto")
os.chdir(r'C:\archivos\proyecto')

if not os.path.exists('Errores'):
    os.makedirs('Errores')

if not os.path.exists('Buenos'):
    os.mkdir('Buenos')

df_hearth = pd.read_csv('indicadores de urgencias2.txt', sep=';')

#df_hearth = pd.read_excel("indicadores de urgencias.xlsx")
```

[51]:
```
df_hearth
# df_hearth.info()
# df_hearth.columns
```

[51]:
```
                 FECHA_LLEGADA             FECHA_TRIAGE  \
0      2023-01-01 01:20:23.853  2023-01-01 01:28:01.847
1      2023-01-01 01:29:46.050  2023-01-01 01:48:03.070
2      2023-01-01 03:15:35.623  2023-01-01 03:23:01.990
3      2023-01-01 05:54:53.563  2023-01-01 06:00:07.943
4      2023-01-01 06:37:27.237  2023-01-01 07:52:31.687
…                          …                         …
82411  2023-09-18 04:44:41.970  2023-09-18 04:53:22.553
82412  2023-09-18 06:17:00.573  2023-09-18 06:28:43.040
82413  2023-09-18 06:21:37.273  2023-09-18 07:00:57.420
82414  2023-09-18 06:25:33.483  2023-09-18 06:42:02.883
82415  2023-09-18 07:14:58.180  2023-09-18 07:30:50.643

                 FECHA_INGRESO           FECHA_ATENCION  \
0      2023-01-01 01:29:41.210  2023-01-01 02:00:07.590
```

```
1      2023-01-01 01:49:40.973  2023-01-01 02:02:53.663
2      2023-01-01 03:23:39.793  2023-01-01 03:30:21.233
3      2023-01-01 06:02:07.320  2023-01-01 06:26:17.050
4      2023-01-01 07:52:37.717  2023-01-01 09:31:15.597
…                          …                        …
82411  2023-09-18 05:05:51.423  2023-09-18 06:09:35.867
82412  2023-09-18 06:35:38.213  2023-09-18 07:40:45.957
82413  2023-09-18 07:16:45.907  2023-09-18 08:27:27.337
82414  2023-09-18 06:51:35.970  2023-09-18 07:28:28.290
82415  2023-09-18 07:34:08.370  2023-09-18 07:49:18.440

      TIEMPO_DGTURNO_A_TRIAGE TIEMPO_TRIAGE_A_INGRESO  \
0                     0:07:38                 0:01:40
1                     0:18:17                 0:01:37
2                     0:07:26                 0:00:38
3                     0:05:14                 0:02:00
4                     1:15:04                 0:00:06
…                       …                       …
82411                 0:08:41                 0:12:29
82412                 0:11:43                 0:06:55
82413                 0:39:20                 0:15:48
82414                 0:16:29                 0:09:33
82415                 0:15:52                 0:03:18

      TIEMPO_INGRESO_A_CONSULTA TIEMPO_TOTAL Tiempo_Minutos_Total  \
0                       0:30:26      0:39:44                39,73
1                       0:13:13      0:33:07                33,12
2                       0:06:42      0:14:46                14,77
3                       0:24:10      0:31:24                31,40
4                       1:38:38      2:53:48               173,80
…                         …          …                        …
82411                   1:03:44      1:24:54                84,90
82412                   1:05:07      1:23:45                83,75
82413                   1:10:42      2:05:50               125,83
82414                   0:36:53      1:02:55                62,92
82415                   0:15:10      0:34:20                34,33

                           CENTRO_ATENCION  …    EDAD_RANGO       SEXO  \
0                       TN - HOSPITAL TUNAL  …      JUVENTUD  MASCULINO
1                     ME - HOSPITAL MEISSEN  …      JUVENTUD   FEMENINO
2      UC - CENTRO DE SALUD SANTA LIBRADA I  …      JUVENTUD  MASCULINO
3      UC - CENTRO DE SALUD SANTA LIBRADA I  …        ADULTO  MASCULINO
4                       TN - HOSPITAL TUNAL  …      JUVENTUD  MASCULINO
…                                         …  …           …         …
82411                 ME - HOSPITAL MEISSEN  …      JUVENTUD   FEMENINO
82412                 ME - HOSPITAL MEISSEN  …  ADULTO MAYOR  MASCULINO
82413                 ME - HOSPITAL MEISSEN  …  ADULTO MAYOR  MASCULINO
```

```
82414             UB - CENTRO DE SALUD USME  …      JUVENTUD  MASCULINO
82415                   TN - HOSPITAL TUNAL  …   ADOLECENCIA   FEMENINO

       RÉGIMEN PACIENTE NOMBRE_ENTIDAD  MEDICO   AÑO MES  DIA_SEMANA  HOUR  \
0             SUBSIDIADO          EPSS34  11065  2023   1     DOMINGO     1
1             SUBSIDIADO          EPSS34   8861  2023   1     DOMINGO     1
2           CONTRIBUTIVO          EPS002   5855  2023   1     DOMINGO     3
3             SUBSIDIADO          EPSS34  11072  2023   1     DOMINGO     5
4             SUBSIDIADO          EPSS34   1239  2023   1     DOMINGO     6
…                    …               …     …   …  ..        …   …
82411       CONTRIBUTIVO          EPS017   7844  2023   9       LUNES     4
82412         SUBSIDIADO          EPSC34   6204  2023   9       LUNES     6
82413       CONTRIBUTIVO          EPS041   9951  2023   9       LUNES     6
82414         SUBSIDIADO          EPSC34   4030  2023   9       LUNES     6
82415         SUBSIDIADO          EPSC34   1239  2023   9       LUNES     7

             Turnos
0       TURNO_NOCHE
1       TURNO_NOCHE
2       TURNO_NOCHE
3       TURNO_NOCHE
4       TURNO_NOCHE
…               …
82411   TURNO_NOCHE
82412   TURNO_NOCHE
82413   TURNO_NOCHE
82414   TURNO_NOCHE
82415  TURNO_MAÑANA

[82416 rows x 23 columns]
```

[11]:
```
# transformar datos
df_hearth['FECHA_INGRESO'].fillna('FECHA_TRIAGE', inplace=True)
df_hearth['FECHA_ATENCION'].fillna('FECHA_INGRESO', inplace=True)
df_hearth['FECHA_TRIAGE'] = pd.to_datetime(df_hearth['FECHA_TRIAGE'],␣
 ↪format='%d/%m/%Y %H:%M')
df_hearth['FECHA_INGRESO'] = pd.to_datetime(df_hearth['FECHA_INGRESO'],␣
 ↪format='%d/%m/%Y %H:%M')
df_hearth['FECHA_LLEGADA'] = pd.to_datetime(df_hearth['FECHA_LLEGADA'],␣
 ↪format='%d/%m/%Y %H:%M')
```

[10]:
```
for i, row in df_hearth.iterrows():
    try:
        df_hearth.at[i, 'FECHA_LLEGADA'] = pd.to_datetime(row['FECHA_LLEGADA'],␣
 ↪format='%d/%m/%Y %H:%M')
    except ValueError:
```

```
        # Manejar el error, por ejemplo, reemplazar el valor incorrecto con una␣
 ↪fecha válida
        df_hearth.at[i, 'FECHA_LLEGADA'] = pd.to_datetime('01/01/2000 00:00',␣
 ↪format='%d/%m/%Y %H:%M')
```

[8]:
```
for i, row in df_hearth.iterrows():
    try:
        df_hearth.at[i, 'FECHA_INGRESO'] = pd.to_datetime(row['FECHA_INGRESO'],␣
 ↪format='%d/%m/%Y %H:%M')
    except ValueError:
        # Manejar el error, por ejemplo, reemplazar el valor incorrecto con una␣
 ↪fecha válida
        df_hearth.at[i, 'FECHA_INGRESO'] = pd.to_datetime('01/01/2000 00:00',␣
 ↪format='%d/%m/%Y %H:%M')
```

[6]:
```
df_hearth['FECHA_CONSULTA'] = pd.to_datetime(df_hearth['FECHA_CONSULTA'],␣
 ↪format='%d/%m/%Y %H:%M')
```

[ ]:
```
# Especifica el formato correcto de las fechas
df_hearth['TIME'] = pd.to_datetime(df_hearth['FECHA_LLEGADA'], format='%d/%m/%Y␣
 ↪%H:%M:%S')

# Extrae el dia
df_hearth['DIA'] = df_hearth['TIME'].dt.day
```

[7]:
```
# Especifica el formato correcto de las fechas
df_hearth['TIME'] = pd.to_datetime(df_hearth['FECHA_LLEGADA'], format='%d/%m/%Y␣
 ↪%H:%M')

# Extrae la hora
df_hearth['HOUR'] = df_hearth['TIME'].dt.hour

# Extrae el mes
df_hearth['MES'] = df_hearth['TIME'].dt.month

# Extrae el dia
df_hearth['DIA'] = df_hearth['TIME'].dt.day

# Extrae el año
df_hearth['AÑO'] = df_hearth['TIME'].dt.year
```

[8]:
```
# crea columna turnos
bins = [0, 6, 12, 18, float("inf")]
labels = ['MADRUGADA', 'MAÑANA', 'TARDE', 'NOCHE']
df_hearth['Turnos'] = pd.cut(df_hearth['HOUR'], bins=bins, labels=labels)
df_hearth['Turnos'].fillna('MADRUGADA', inplace=True)
```

[ ]:
```
# mostrar las columnas que quiero
'''
df1 = df_hearth.head(200)
numeric_columns = ['FECHA_LLEGADA', 'Turnos', 'HOUR']
df2 = df1[numeric_columns]
df2.to_csv('ind_urgencias.txt', index=False)
'''
```

[9]:
```
# Define la fecha que deseas buscar
fecha_a_buscar = pd.to_datetime('01/01/2000 00:00')

# Encuentra las filas que contienen la fecha deseada en 'FECHA_CONSULTA'
filas_a_actualizar = df_hearth[df_hearth['FECHA_CONSULTA'] == fecha_a_buscar]

# Reemplaza la fecha en 'FECHA_CONSULTA' con la fecha correspondiente en␣
 ↪'FECHA_INGRESO'
df_hearth.loc[filas_a_actualizar.index, 'FECHA_CONSULTA'] = df_hearth.
 ↪loc[filas_a_actualizar.index, 'FECHA_INGRESO']
```

[10]:
```
# Calcula la diferencia de tiempo y crea una nueva columna 'Tiempo_Consulta' en␣
 ↪minutos
df_hearth['Tiempo_Triage'] = (df_hearth['FECHA_TRIAGE'] -␣
 ↪df_hearth['FECHA_LLEGADA']).dt.total_seconds() / 60
df_hearth['Tiempo_Ingreso'] = (df_hearth['FECHA_INGRESO'] -␣
 ↪df_hearth['FECHA_TRIAGE']).dt.total_seconds() / 60
df_hearth['Tiempo_Consulta'] = (df_hearth['FECHA_CONSULTA'] -␣
 ↪df_hearth['FECHA_INGRESO']).dt.total_seconds() / 60
df_hearth['Tiempo_Minutos_Total'] = (df_hearth['Tiempo_Triage'] +␣
 ↪df_hearth['Tiempo_Ingreso'] + df_hearth['Tiempo_Consulta'])
```

[53]:
```
# trasnformar datos que no requiero
# df_hearth
df_hearth['CENTRO_ATENCION'] = df_hearth['CENTRO_ATENCION'].str.slice(0, 2)
df_hearth['CLASIFICACION_TRIAGE'] = df_hearth['CLASIFICACION_TRIAGE'].str.
 ↪slice(0, 1)
# df_hearth['Turnos'] = df_hearth['Turnos'].str.slice(3, 0)
df_hearth['Turnos'] = df_hearth['Turnos'].str[6:]
# df_hearth['PACIENTE_EDAD'] = df_hearth['PACIENTE_EDAD'].str.slice(0, 3)
# df_hearth['PACIENTE_EDAD'] = df_hearth['PACIENTE_EDAD'].apply(lambda x: re.
 ↪sub(r'\D', '', str(x)))
```

[12]:
```
df_hearth['DIA_SEMANA'] = df_hearth['TIME'].dt.day_name()
```

[46]:
```
df_hearth['Turnos'] = df_hearth['Turnos'].astype('category')
df_hearth['DIA_SEMANA'] = df_hearth['DIA_SEMANA'].astype('category')
```

[56]:
```
df_hearth.to_csv('ind_urgencias_final_2023.txt', sep=';', index=False)
```

[57]:
```
df_hearth
```

[57]:

	FECHA_LLEGADA	FECHA_TRIAGE
0	2023-01-01 01:20:23.853	2023-01-01 01:28:01.847
1	2023-01-01 01:29:46.050	2023-01-01 01:48:03.070
2	2023-01-01 03:15:35.623	2023-01-01 03:23:01.990
3	2023-01-01 05:54:53.563	2023-01-01 06:00:07.943
4	2023-01-01 06:37:27.237	2023-01-01 07:52:31.687
…	…	…
82411	2023-09-18 04:44:41.970	2023-09-18 04:53:22.553
82412	2023-09-18 06:17:00.573	2023-09-18 06:28:43.040
82413	2023-09-18 06:21:37.273	2023-09-18 07:00:57.420
82414	2023-09-18 06:25:33.483	2023-09-18 06:42:02.883
82415	2023-09-18 07:14:58.180	2023-09-18 07:30:50.643

	FECHA_INGRESO	FECHA_ATENCION
0	2023-01-01 01:29:41.210	2023-01-01 02:00:07.590
1	2023-01-01 01:49:40.973	2023-01-01 02:02:53.663
2	2023-01-01 03:23:39.793	2023-01-01 03:30:21.233
3	2023-01-01 06:02:07.320	2023-01-01 06:26:17.050
4	2023-01-01 07:52:37.717	2023-01-01 09:31:15.597
…	…	…
82411	2023-09-18 05:05:51.423	2023-09-18 06:09:35.867
82412	2023-09-18 06:35:38.213	2023-09-18 07:40:45.957
82413	2023-09-18 07:16:45.907	2023-09-18 08:27:27.337
82414	2023-09-18 06:51:35.970	2023-09-18 07:28:28.290
82415	2023-09-18 07:34:08.370	2023-09-18 07:49:18.440

	TIEMPO_DGTURNO_A_TRIAGE	TIEMPO_TRIAGE_A_INGRESO
0	0:07:38	0:01:40
1	0:18:17	0:01:37
2	0:07:26	0:00:38
3	0:05:14	0:02:00
4	1:15:04	0:00:06
…	…	…
82411	0:08:41	0:12:29
82412	0:11:43	0:06:55
82413	0:39:20	0:15:48
82414	0:16:29	0:09:33
82415	0:15:52	0:03:18

	TIEMPO_INGRESO_A_CONSULTA	TIEMPO_TOTAL	Tiempo_Minutos_Total
0	0:30:26	0:39:44	39,73
1	0:13:13	0:33:07	33,12
2	0:06:42	0:14:46	14,77

```
3                      0:24:10      0:31:24               31,40
4                      1:38:38      2:53:48              173,80
…                        …          …                      …
82411                  1:03:44      1:24:54               84,90
82412                  1:05:07      1:23:45               83,75
82413                  1:10:42      2:05:50              125,83
82414                  0:36:53      1:02:55               62,92
82415                  0:15:10      0:34:20               34,33

      CENTRO_ATENCION  …    EDAD_RANGO       SEXO  RÉGIMEN PACIENTE  \
0                  TN  …       JUVENTUD  MASCULINO        SUBSIDIADO
1                  ME  …       JUVENTUD   FEMENINO        SUBSIDIADO
2                  UC  …       JUVENTUD  MASCULINO      CONTRIBUTIVO
3                  UC  …         ADULTO  MASCULINO        SUBSIDIADO
4                  TN  …       JUVENTUD  MASCULINO        SUBSIDIADO
…                …  …            …          …                 …
82411              ME  …       JUVENTUD   FEMENINO      CONTRIBUTIVO
82412              ME  …  ADULTO MAYOR  MASCULINO        SUBSIDIADO
82413              ME  …  ADULTO MAYOR  MASCULINO      CONTRIBUTIVO
82414              UB  …       JUVENTUD  MASCULINO        SUBSIDIADO
82415              TN  …    ADOLECENCIA   FEMENINO        SUBSIDIADO

      NOMBRE_ENTIDAD  MEDICO   AÑO MES  DIA_SEMANA  HOUR  Turnos
0              EPSS34  11065  2023   1     DOMINGO     1   NOCHE
1              EPSS34   8861  2023   1     DOMINGO     1   NOCHE
2              EPS002   5855  2023   1     DOMINGO     3   NOCHE
3              EPSS34  11072  2023   1     DOMINGO     5   NOCHE
4              EPSS34   1239  2023   1     DOMINGO     6   NOCHE
…                …    …   …  ..          …    …     …
82411          EPS017   7844  2023   9       LUNES     4   NOCHE
82412          EPSC34   6204  2023   9       LUNES     6   NOCHE
82413          EPS041   9951  2023   9       LUNES     6   NOCHE
82414          EPSC34   4030  2023   9       LUNES     6   NOCHE
82415          EPSC34   1239  2023   9       LUNES     7  MAÑANA

[82416 rows x 23 columns]
```

[17]:
```
df_hearth
```

[17]:
```
        TIEMPO_LLEGADA_A_TRIAGE TIEMPO_TRIAGE_A_INGRESO  \
0                       0:00:00                0:16:13
1                       0:00:00                0:08:06
2                       0:00:00                0:33:41
3                       0:00:00                0:17:52
4                       0:00:00                0:13:57
…                         …                      …
1510868                 0:06:39                0:03:20
```

```
1510869                 1:17:38                 0:01:19
1510870                 0:24:11                 0:00:32
1510871                 0:05:44                 0:07:40
1510872                 0:09:18                 0:01:08

        TIEMPO_INGRESO_A_FOLIO TIEMPO_TOTAL       FECHA_LLEGADA  \
0                      0:08:47      0:25:00 2016-08-02 07:19:00
1                      0:01:27      0:09:33 2016-08-02 07:58:00
2                      0:04:12      0:37:53 2016-08-03 09:10:00
3                      0:02:22      0:20:14 2016-08-03 13:05:00
4                      0:21:41      0:35:38 2016-08-01 08:35:00
…                        …          …                 …
1510868                0:00:00      0:00:00 2023-09-14 19:48:00
1510869                0:13:29      1:32:26 2023-09-14 18:40:00
1510870                0:01:16      0:25:59 2023-09-14 19:37:00
1510871                0:00:00      0:00:00 2023-09-14 19:50:00
1510872                0:00:00      0:00:00 2023-09-14 19:55:00

               FECHA_TRIAGE       FECHA_INGRESO      FECHA_CONSULTA  \
0       2016-08-02 07:19:00 2016-08-02 07:36:00 2016-08-02 07:45:00
1       2016-08-02 07:58:00 2016-08-02 08:07:00 2016-08-02 08:09:00
2       2016-08-03 09:10:00 2016-08-03 09:44:00 2016-08-03 09:48:00
3       2016-08-03 13:05:00 2016-08-03 13:23:00 2016-08-03 13:26:00
4       2016-08-01 08:35:00 2016-08-01 08:49:00 2016-08-01 09:11:00
…                       …                   …                 …
1510868 2023-09-14 19:55:00 2023-09-14 19:58:00 2023-09-14 20:02:00
1510869 2023-09-14 19:57:00 2023-09-14 19:59:00 2023-09-14 20:00:00
1510870 2023-09-14 20:02:00 2023-09-14 20:02:00 2023-09-14 20:03:00
1510871 2023-09-14 19:56:00 2023-09-14 20:03:00 2023-09-14 20:11:00
1510872 2023-09-14 20:04:00 2023-09-14 20:05:00 2023-09-14 20:06:00

        CENTRO_ATENCION CLASIFICACION_TRIAGE  … HOUR MES DIA   AÑO  Turnos  \
0                    VA                    3  …    7   8   2  2016  MAÑANA
1                    VA                    3  …    7   8   2  2016  MAÑANA
2                    VA                    3  …    9   8   3  2016  MAÑANA
3                    VC                    3  …   13   8   3  2016   TARDE
4                    ME                    3  …    8   8   1  2016  MAÑANA
…                 …                      …  …  …  ..  ..   …     …
1510868              UC                    3  …   19   9  14  2023   NOCHE
1510869              JT                    3  …   18   9  14  2023   TARDE
1510870              ME                    2  …   19   9  14  2023   NOCHE
1510871              UC                    3  …   19   9  14  2023   NOCHE
1510872              ME                    3  …   19   9  14  2023   NOCHE

        Tiempo_Triage Tiempo_Ingreso Tiempo_Consulta  Tiempo_Minutos_Total  \
0                 0.0           17.0             9.0                  26.0
1                 0.0            9.0             2.0                  11.0
```

```
2                0.0          34.0            4.0                38.0
3                0.0          18.0            3.0                21.0
4                0.0          14.0           22.0                36.0
…                …            …               …                  …
1510868          7.0           3.0            4.0                14.0
1510869         77.0           2.0            1.0                80.0
1510870         25.0           0.0            1.0                26.0
1510871          6.0           7.0            8.0                21.0
1510872          9.0           1.0            1.0                11.0

          DIA_SEMANA
0            Tuesday
1            Tuesday
2          Wednesday
3          Wednesday
4             Monday
…               …
1510868     Thursday
1510869     Thursday
1510870     Thursday
1510871     Thursday
1510872     Thursday

[1510873 rows x 28 columns]
```

[25]:
```
# filtrado por año
FILTRO = 2023
df_filtrado = df_hearth[df_hearth['AÑO'] == FILTRO]
df_filtrado.to_csv('ind_urgencias_final_2023.txt', sep=';', index=False)
```

[26]:
```
df_filtrado
```

[26]:
```
         TIEMPO_LLEGADA_A_TRIAGE TIEMPO_TRIAGE_A_INGRESO  \
1373529                 0:00:00                 0:11:26
1373530                 0:00:00                 0:00:20
1373531                 0:15:58                 0:01:01
1373532                 0:00:00                 0:00:00
1373533                 0:00:00                 0:05:00
…                         …                     …
1510868                 0:06:39                 0:03:20
1510869                 1:17:38                 0:01:19
1510870                 0:24:11                 0:00:32
1510871                 0:05:44                 0:07:40
1510872                 0:09:18                 0:01:08

         TIEMPO_INGRESO_A_FOLIO TIEMPO_TOTAL       FECHA_LLEGADA  \
1373529                0:05:59      0:17:25 2023-01-01 00:04:00
```

```
1373530               0:29:25      0:29:45 2023-01-01 00:16:00
1373531               0:27:06      0:44:05 2023-01-01 00:05:00
1373532               0:00:00      0:00:00 2023-01-01 00:24:00
1373533               0:01:01      0:06:01 2023-01-01 00:24:00
…                       …          …                 …
1510868               0:00:00      0:00:00 2023-09-14 19:48:00
1510869               0:13:29      1:32:26 2023-09-14 18:40:00
1510870               0:01:16      0:25:59 2023-09-14 19:37:00
1510871               0:00:00      0:00:00 2023-09-14 19:50:00
1510872               0:00:00      0:00:00 2023-09-14 19:55:00

              FECHA_TRIAGE       FECHA_INGRESO      FECHA_CONSULTA  \
1373529 2023-01-01 00:04:00 2023-01-01 00:15:00 2023-01-01 00:27:00
1373530 2023-01-01 00:16:00 2023-01-01 00:17:00 2023-01-01 00:17:00
1373531 2023-01-01 00:21:00 2023-01-01 00:22:00 2023-01-01 00:23:00
1373532 2023-01-01 00:24:00 2023-01-01 00:24:00 2023-01-01 00:24:00
1373533 2023-01-01 00:24:00 2023-01-01 00:29:00 2023-01-01 00:34:00
…                       …                   …                 …
1510868 2023-09-14 19:55:00 2023-09-14 19:58:00 2023-09-14 20:02:00
1510869 2023-09-14 19:57:00 2023-09-14 19:59:00 2023-09-14 20:00:00
1510870 2023-09-14 20:02:00 2023-09-14 20:02:00 2023-09-14 20:03:00
1510871 2023-09-14 19:56:00 2023-09-14 20:03:00 2023-09-14 20:11:00
1510872 2023-09-14 20:04:00 2023-09-14 20:05:00 2023-09-14 20:06:00

        CENTRO_ATENCION CLASIFICACION_TRIAGE  … HOUR MES DIA   AÑO  \
1373529              JT                    3  …    0   1   1  2023
1373530              TN                    3  …    0   1   1  2023
1373531              TN                    3  …    0   1   1  2023
1373532              VB                    3  …    0   1   1  2023
1373533              VC                    3  …    0   1   1  2023
…                 …                    …  …  …   ..  ..   …
1510868              UC                    3  …   19   9  14  2023
1510869              JT                    3  …   18   9  14  2023
1510870              ME                    2  …   19   9  14  2023
1510871              UC                    3  …   19   9  14  2023
1510872              ME                    3  …   19   9  14  2023

           Turnos Tiempo_Triage Tiempo_Ingreso Tiempo_Consulta  \
1373529  MADRUGADA           0.0           11.0            12.0
1373530  MADRUGADA           0.0            1.0             0.0
1373531  MADRUGADA          16.0            1.0             1.0
1373532  MADRUGADA           0.0            0.0             0.0
1373533  MADRUGADA           0.0            5.0             5.0
…            …           …             …              …
1510868      NOCHE           7.0            3.0             4.0
1510869      TARDE          77.0            2.0             1.0
1510870      NOCHE          25.0            0.0             1.0
```

```
1510871      NOCHE           6.0            7.0             8.0
1510872      NOCHE           9.0            1.0             1.0

         Tiempo_Minutos_Total  DIA_SEMANA
1373529                  23.0      Sunday
1373530                   1.0      Sunday
1373531                  18.0      Sunday
1373532                   0.0      Sunday
1373533                  10.0      Sunday
…                         …          …
1510868                  14.0    Thursday
1510869                  80.0    Thursday
1510870                  26.0    Thursday
1510871                  21.0    Thursday
1510872                  11.0    Thursday

[137335 rows x 28 columns]
```

[15]:
```
df_filtrado.head(5)
```

[15]:
```
Empty DataFrame
Columns: [TIEMPO_LLEGADA_A_TRIAGE, TIEMPO_TRIAGE_A_INGRESO,
TIEMPO_INGRESO_A_FOLIO, TIEMPO_TOTAL, FECHA_LLEGADA, FECHA_TRIAGE,
FECHA_INGRESO, FECHA_CONSULTA, CENTRO_ATENCION, CLASIFICACION_TRIAGE,
PACIENTE_#_DOCUMENTO, PACIENTE_EDAD, EDAD_RANGO, SEXO, RÉGIMEN PACIENTE,
NOMBRE_ENTIDAD, DIAGNOSTICO, TIME, HOUR, MES, DIA, AÑO, Turnos, Tiempo_Triage,
Tiempo_Ingreso, Tiempo_Consulta, Tiempo_Minutos_Total, DIA_SEMANA]
Index: []

[0 rows x 28 columns]
```

[97]:
```
# Definir una lista de archivos y estructuras

archivos_y_estructuras = [
(f'ind_urgencias_final.txt',{ 'TIEMPO_LLEGADA_A_TRIAGE': str,␣
 ↪'TIEMPO_TRIAGE_A_INGRESO': str,
       'TIEMPO_INGRESO_A_FOLIO': str, 'TIEMPO_TOTAL': str, 'FECHA_LLEGADA': str,
       'FECHA_TRIAGE': str, 'FECHA_INGRESO': str, 'FECHA_CONSULTA': str,␣
 ↪'CENTRO_ATENCION': str,
       'CLASIFICACION_TRIAGE': str, 'PACIENTE_#_DOCUMENTO': str,␣
 ↪'PACIENTE_EDAD': str,
       'EDAD_RANGO': str, 'SEXO': str, 'RÉGIMEN PACIENTE': str,␣
 ↪'NOMBRE_ENTIDAD': str,
       'NOM_TIPO_HISTORIA': str, 'DIAGNOSTICO': str, 'NOMBRE DX': str, 'TIME':␣
 ↪str, 'HOUR': str, 'MES': str,
       'DIA': str, 'Turnos': str, 'Tiempo_Triague': str, 'Tiempo_Ingreso': str
})
```

```
                ]

for archivo, estructura in archivos_y_estructuras:
    df_buenas = []
    df_errores = []

    with open(archivo, 'r', encoding='utf-8') as file:
        # Leer la primera línea (encabezado) y omitirla
        header = next(file)
        for line in file:
            try:
                observacion = ''  # Inicializa la observación como vacía
                data = line.strip().split(';')
                if len(data) != len(estructura):
                    observacion += "La estrutura no cuadra con la original. Hay␣
 ↪caracteres malos "
                    data[-1] = observacion.strip()
                    df_errores.append(data)
                else:
                    observacion = ''  # Inicializa la observación como vacía
                    registro = pd.Series(data, index=estructura.keys())
                    if not pd.notna(registro['FECHA_CONSULTA']) or not␣
 ↪registro['FECHA_CONSULTA'].strip():
                        observacion += "Fecha consulta no puede estar vacía. "
                        data[-1] = observacion.strip()
                        df_errores.append(data)
                    elif not pd.notna(registro['FECHA_INGRESO']) or not␣
 ↪registro['FECHA_INGRESO'].strip():
                        observacion += "Fecha registro no puede estar vacía. "
                        data[-1] = observacion.strip()
                        df_errores.append(data)
                    else:
                        df_buenas.append(data)
            except Exception as e:
                df_errores.append([str(e), ''])  # Agrega una columna vacía␣
 ↪para la observación de errores

    df_errores_df = pd.DataFrame({'Errores': df_errores})
    df_buenas_df = pd.DataFrame({'Buenas': df_buenas})

    # Guardar archivos de errores y buenos
    nombre_base = archivo.split('.')[0]
    df_errores_df.to_csv(f'Validar/Errores/{nombre_base}_errores.txt',␣
 ↪index=False)
    df_buenas_df.to_csv(f'Validar/Buenos/{nombre_base}_buenos.txt', index=False)
```