COVID_data_cleanning

March 4, 2025

1 Covid data analysis project

1.1 Team members:

- Coconi
- Sánchez
- Cortés

[ ]:

```

```

[3]:

```
import pandas as pd

# First, lets take a look to the data assets available.
covid_df = pd.read_csv("./assets/Datos COVID/220720COVID19MEXICO.csv",␣
 ↪encoding="latin1", on_bad_lines="warn")

covid_df.head()
```

```
/var/folders/cg/l549l5jn7ql8v5_67ml6_p2m0000gn/T/ipykernel_28358/4023574627.py:4
: ParserWarning: Skipping line 16734185: expected 40 fields, saw 64
Skipping line 16734297: expected 40 fields, saw 62
Skipping line 16736952: expected 40 fields, saw 52
Skipping line 16740173: expected 40 fields, saw 50
Skipping line 16743964: expected 40 fields, saw 54

  covid_df = pd.read_csv("./assets/Datos COVID/220720COVID19MEXICO.csv",
encoding="latin1", on_bad_lines="warn")
/var/folders/cg/l549l5jn7ql8v5_67ml6_p2m0000gn/T/ipykernel_28358/4023574627.py:4
: ParserWarning: Skipping line 16747188: expected 40 fields, saw 58
Skipping line 16749845: expected 40 fields, saw 41
Skipping line 16753065: expected 40 fields, saw 59
Skipping line 16756285: expected 40 fields, saw 45
Skipping line 16759504: expected 40 fields, saw 57
Skipping line 16759956: expected 40 fields, saw 66
Skipping line 16759957: expected 40 fields, saw 52

  covid_df = pd.read_csv("./assets/Datos COVID/220720COVID19MEXICO.csv",
encoding="latin1", on_bad_lines="warn")
/var/folders/cg/l549l5jn7ql8v5_67ml6_p2m0000gn/T/ipykernel_28358/4023574627.py:4
```

```
: ParserWarning: Skipping line 16762725: expected 40 fields, saw 59
Skipping line 16762775: expected 40 fields, saw 63
Skipping line 16765945: expected 40 fields, saw 47

  covid_df = pd.read_csv("./assets/Datos COVID/220720COVID19MEXICO.csv",
encoding="latin1", on_bad_lines="warn")
/var/folders/cg/l54915jn7ql8v5_67ml6_p2m0000gn/T/ipykernel_28358/4023574627.py:4
: DtypeWarning: Columns (2,6,8,25,26,27,38) have mixed types. Specify dtype
option on import or set low_memory=False.
  covid_df = pd.read_csv("./assets/Datos COVID/220720COVID19MEXICO.csv",
encoding="latin1", on_bad_lines="warn")
```

[3]:

```
  FECHA_ACTUALIZACION ID_REGISTRO ORIGEN  SECTOR  ENTIDAD_UM  SEXO  \
0          2022-07-20      z3bf80      2      12         8.0   2.0
1          2022-07-20      z1e370      1      12        14.0   1.0
2          2022-07-20      zze974      1       6        24.0   1.0
3          2022-07-20      zz7067      1      12         9.0   2.0
4          2022-07-20      z1da1e      1      12         1.0   2.0

  ENTIDAD_NAC  ENTIDAD_RES MUNICIPIO_RES  TIPO_PACIENTE  … OTRO_CASO  \
0           8          8.0            37            1.0  …       2.0
1          14         14.0            85            1.0  …       2.0
2          24         24.0            35            1.0  …       1.0
3           9          9.0             7            1.0  …       2.0
4           1          1.0             1            1.0  …       1.0

  TOMA_MUESTRA_LAB RESULTADO_LAB  TOMA_MUESTRA_ANTIGENO  RESULTADO_ANTIGENO  \
0              1.0           1.0                    2.0                97.0
1              1.0           2.0                    2.0                97.0
2              1.0           2.0                    2.0                97.0
3              1.0           2.0                    2.0                97.0
4              1.0           2.0                    2.0                97.0

   CLASIFICACION_FINAL  MIGRANTE  PAIS_NACIONALIDAD  PAIS_ORIGEN   UCI
0                  3.0      99.0             MÃ©xico           97  97.0
1                  7.0      99.0             MÃ©xico           97  97.0
2                  7.0      99.0             MÃ©xico           97  97.0
3                  7.0      99.0             MÃ©xico           97  97.0
4                  7.0      99.0             MÃ©xico           97  97.0

[5 rows x 40 columns]
```

[4]:

```
covid_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17311026 entries, 0 to 17311025
Data columns (total 40 columns):
 #   Column                 Dtype
```

```
---  ------                 -----
 0   FECHA_ACTUALIZACION    object
 1   ID_REGISTRO            object
 2   ORIGEN                 object
 3   SECTOR                 int64
 4   ENTIDAD_UM             float64
 5   SEXO                   float64
 6   ENTIDAD_NAC            object
 7   ENTIDAD_RES            float64
 8   MUNICIPIO_RES          object
 9   TIPO_PACIENTE          float64
 10  FECHA_INGRESO          object
 11  FECHA_SINTOMAS         object
 12  FECHA_DEF              object
 13  INTUBADO               float64
 14  NEUMONIA               float64
 15  EDAD                   float64
 16  NACIONALIDAD           float64
 17  EMBARAZO               float64
 18  HABLA_LENGUA_INDIG     float64
 19  INDIGENA               float64
 20  DIABETES               float64
 21  EPOC                   float64
 22  ASMA                   float64
 23  INMUSUPR               float64
 24  HIPERTENSION           float64
 25  OTRA_COM               object
 26  CARDIOVASCULAR         object
 27  OBESIDAD               object
 28  RENAL_CRONICA          float64
 29  TABAQUISMO             float64
 30  OTRO_CASO              float64
 31  TOMA_MUESTRA_LAB       float64
 32  RESULTADO_LAB          float64
 33  TOMA_MUESTRA_ANTIGENO  float64
 34  RESULTADO_ANTIGENO     float64
 35  CLASIFICACION_FINAL    float64
 36  MIGRANTE               float64
 37  PAIS_NACIONALIDAD      object
 38  PAIS_ORIGEN            object
 39  UCI                    float64
dtypes: float64(26), int64(1), object(13)
memory usage: 5.2+ GB
```

[5]:

```
covid_df.columns
```

```
[5]: Index(['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'],
           dtype='object')
```

```
[6]: numeric_columns = covid_df.select_dtypes(include=['int64', 'float64']).columns
     numeric_df = covid_df[numeric_columns]
     numeric_df.describe()
```

```
[6]:              SECTOR    ENTIDAD_UM          SEXO   ENTIDAD_RES  TIPO_PACIENTE  \
     count  1.731103e+07  1.731102e+07  1.731102e+07  1.731102e+07   1.731102e+07
     mean   8.730032e+00  1.423885e+01  1.463256e+00  1.451560e+01   1.073313e+00
     std    3.826896e+00  7.783762e+00  4.994920e-01  7.715083e+00   2.616575e-01
     min    1.000000e+00  1.000000e+00  1.000000e+00  1.000000e+00   1.000000e+00
     25%    4.000000e+00  9.000000e+00  1.000000e+00  9.000000e+00   1.000000e+00
     50%    1.200000e+01  1.100000e+01  1.000000e+00  1.200000e+01   1.000000e+00
     75%    1.200000e+01  2.000000e+01  2.000000e+00  2.000000e+01   1.000000e+00
     max    9.900000e+01  7.300000e+01  9.700000e+01  9.700000e+01   9.700000e+01

                INTUBADO      NEUMONIA          EDAD  NACIONALIDAD      EMBARAZO  \
     count  1.731102e+07  1.731102e+07  1.731102e+07  1.731102e+07  1.731102e+07
     mean   9.010602e+01  2.919721e+00  3.841132e+01  1.007196e+00  4.640194e+01
     std    2.466175e+01  9.631083e+00  1.713820e+01  8.452546e-02  4.741358e+01
     min    1.000000e+00  1.000000e+00  0.000000e+00  1.000000e+00  1.000000e+00
     25%    9.700000e+01  2.000000e+00  2.600000e+01  1.000000e+00  2.000000e+00
     50%    9.700000e+01  2.000000e+00  3.700000e+01  1.000000e+00  2.000000e+00
     75%    9.700000e+01  2.000000e+00  5.000000e+01  1.000000e+00  9.700000e+01
     max    9.900000e+01  9.900000e+01  2.660000e+02  2.000000e+00  9.900000e+01

            …  RENAL_CRONICA    TABAQUISMO     OTRO_CASO  TOMA_MUESTRA_LAB  \
     count  …   1.731102e+07  1.731102e+07  1.731102e+07      1.731102e+07
     mean   …   2.473940e+00  2.429364e+00  4.981694e+00      1.660153e+00
     std    …   6.796166e+00  6.856243e+00  1.768579e+01      4.736572e-01
     min    …   1.000000e+00  1.000000e+00  1.000000e+00      1.000000e+00
     25%    …   2.000000e+00  2.000000e+00  1.000000e+00      1.000000e+00
     50%    …   2.000000e+00  2.000000e+00  2.000000e+00      2.000000e+00
     75%    …   2.000000e+00  2.000000e+00  2.000000e+00      2.000000e+00
     max    …   9.800000e+01  9.800000e+01  9.900000e+01      2.000000e+00

            RESULTADO_LAB  TOMA_MUESTRA_ANTIGENO  RESULTADO_ANTIGENO  \
     count   1.731102e+07           1.731102e+07        1.731102e+07
```

```
mean    6.461132e+01           1.324281e+00        3.257903e+01
std     4.514311e+01           4.681057e-01        4.462946e+01
min     1.000000e+00           1.000000e+00        1.000000e+00
25%     2.000000e+00           1.000000e+00        2.000000e+00
50%     9.700000e+01           1.000000e+00        2.000000e+00
75%     9.700000e+01           2.000000e+00        9.700000e+01
max     9.700000e+01           2.000000e+00        9.700000e+01

       CLASIFICACION_FINAL      MIGRANTE          UCI
count         1.731102e+07  1.731102e+07  1.731102e+07
mean          5.401664e+00  9.835919e+01  9.010827e+01
std           1.988091e+00  7.866601e+00  2.465428e+01
min           1.000000e+00  1.000000e+00  1.000000e+00
25%           3.000000e+00  9.900000e+01  9.700000e+01
50%           7.000000e+00  9.900000e+01  9.700000e+01
75%           7.000000e+00  9.900000e+01  9.700000e+01
max           7.000000e+00  9.900000e+01  9.900000e+01

[8 rows x 27 columns]
```

[7]:

```
df_numeric = covid_df[['SECTOR', 'SECTOR', 'ENTIDAD_UM', 'SEXO', 'ENTIDAD_RES',␣
 ↪'TIPO_PACIENTE', 'INTUBADO', 'NEUMONIA', 'EDAD', 'NACIONALIDAD', 'EMBARAZO',␣
 ↪'HABLA_LENGUA_INDIG', 'INDIGENA']]
```

[8]:

```
tendencia_central = numeric_df.describe().applymap(lambda x: f"{x:0.3f}")
tendencia_central
```

```
/var/folders/cg/l54915jn7ql8v5_67ml6_p2m0000gn/T/ipykernel_28358/3745017783.py:1
: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map
instead.
  tendencia_central = numeric_df.describe().applymap(lambda x: f"{x:0.3f}")
```

[8]:

```
             SECTOR    ENTIDAD_UM          SEXO   ENTIDAD_RES TIPO_PACIENTE  \
count  17311026.000  17311024.000  17311024.000  17311024.000  17311023.000
mean          8.730        14.239         1.463        14.516         1.073
std           3.827         7.784         0.499         7.715         0.262
min           1.000         1.000         1.000         1.000         1.000
25%           4.000         9.000         1.000         9.000         1.000
50%          12.000        11.000         1.000        12.000         1.000
75%          12.000        20.000         2.000        20.000         1.000
max          99.000        73.000        97.000        97.000        97.000

           INTUBADO      NEUMONIA          EDAD  NACIONALIDAD      EMBARAZO  \
count  17311021.000  17311021.000  17311020.000  17311020.000  17311020.000
mean         90.106         2.920        38.411         1.007        46.402
std          24.662         9.631        17.138         0.085        47.414
min           1.000         1.000         0.000         1.000         1.000
25%          97.000         2.000        26.000         1.000         2.000
```

```
50%          97.000        2.000       37.000        1.000        2.000
75%          97.000        2.000       50.000        1.000       97.000
max          99.000       99.000      266.000        2.000       99.000

       … RENAL_CRONICA    TABAQUISMO     OTRO_CASO TOMA_MUESTRA_LAB  \
count  …  17311019.000  17311018.000  17311015.000      17311015.000
mean   …         2.474         2.429         4.982             1.660
std    …         6.796         6.856        17.686             0.474
min    …         1.000         1.000         1.000             1.000
25%    …         2.000         2.000         1.000             1.000
50%    …         2.000         2.000         2.000             2.000
75%    …         2.000         2.000         2.000             2.000
max    …        98.000        98.000        99.000             2.000

      RESULTADO_LAB TOMA_MUESTRA_ANTIGENO RESULTADO_ANTIGENO  \
count  17311015.000          17311015.000        17311015.000
mean         64.611                 1.324              32.579
std          45.143                 0.468              44.629
min           1.000                 1.000               1.000
25%           2.000                 1.000               2.000
50%          97.000                 1.000               2.000
75%          97.000                 2.000              97.000
max          97.000                 2.000              97.000

      CLASIFICACION_FINAL      MIGRANTE           UCI
count        17311015.000  17311015.000  17311015.000
mean                5.402        98.359        90.108
std                 1.988         7.867        24.654
min                 1.000         1.000         1.000
25%                 3.000        99.000        97.000
50%                 7.000        99.000        97.000
75%                 7.000        99.000        97.000
max                 7.000        99.000        99.000

[8 rows x 27 columns]
```

[ ]:

```

```

[9]:

```
df_numeric.info()
```

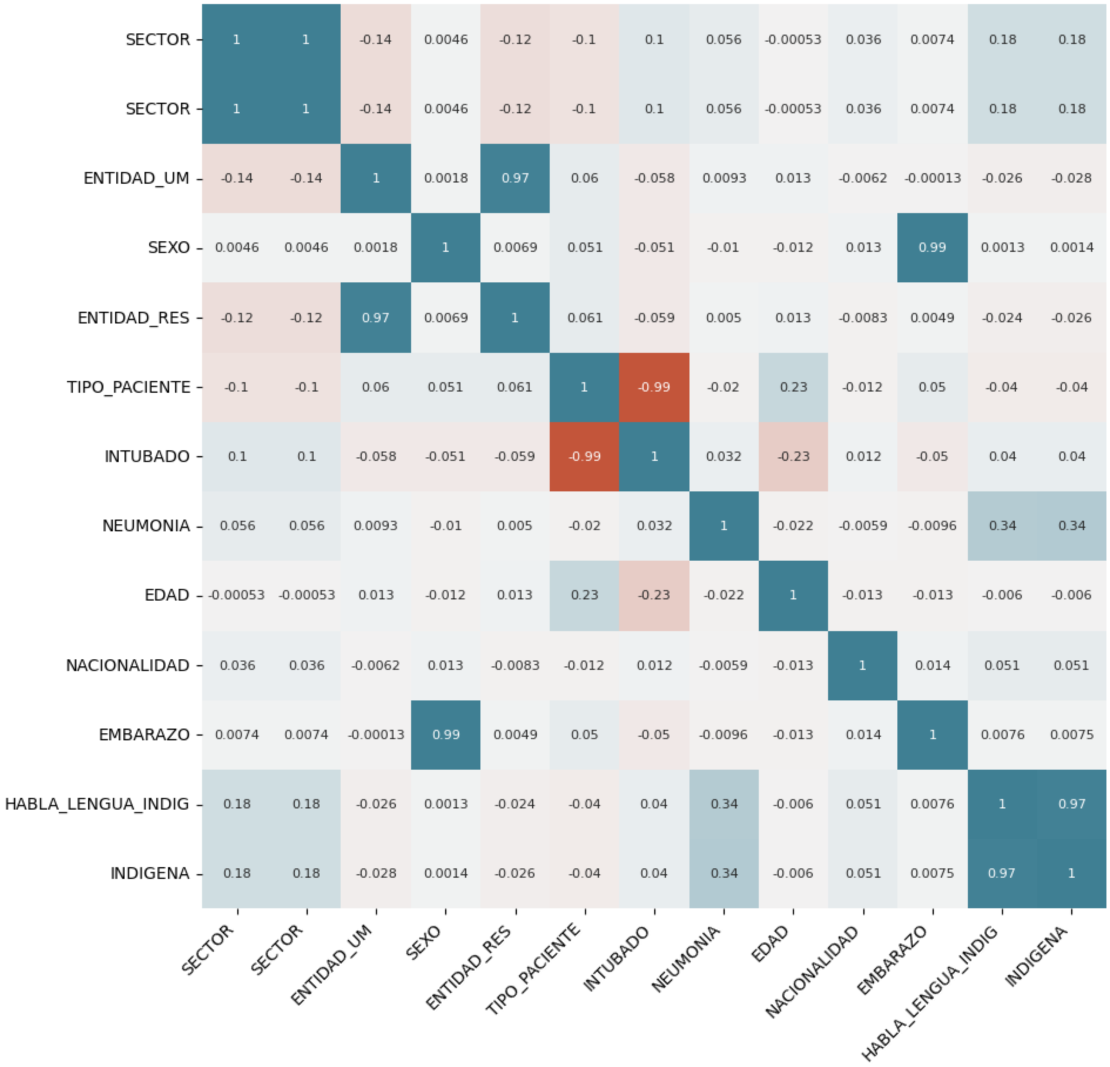
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17311026 entries, 0 to 17311025
Data columns (total 13 columns):
 #   Column             Dtype
---  ------             -----
 0   SECTOR             int64
 1   SECTOR             int64
 2   ENTIDAD_UM         float64
 3   SEXO               float64
 4   ENTIDAD_RES        float64
 5   TIPO_PACIENTE      float64
 6   INTUBADO           float64
 7   NEUMONIA           float64
 8   EDAD               float64
 9   NACIONALIDAD       float64
 10  EMBARAZO           float64
 11  HABLA_LENGUA_INDIG  float64
 12  INDIGENA           float64
dtypes: float64(11), int64(2)
memory usage: 1.7 GB
```

[10]:

```python
corr_matrix = df_numeric.corr(method='pearson')

# Print corr matrix as a pretty chart of big size
import matplotlib.pyplot as plt
import seaborn as sns


fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(10, 10))
sns.heatmap(corr_matrix,annot=True,cbar=False,annot_kws = {"size":␣
 ↪8},vmin=-1,vmax=1,center=0,
    cmap=sns.diverging_palette(20, 220, n=200), square=True,ax=ax)
ax.set_xticklabels(ax.get_xticklabels(),rotation = 45,horizontalalignment =␣
 ↪'right',)
ax.tick_params(labelsize = 10)
```

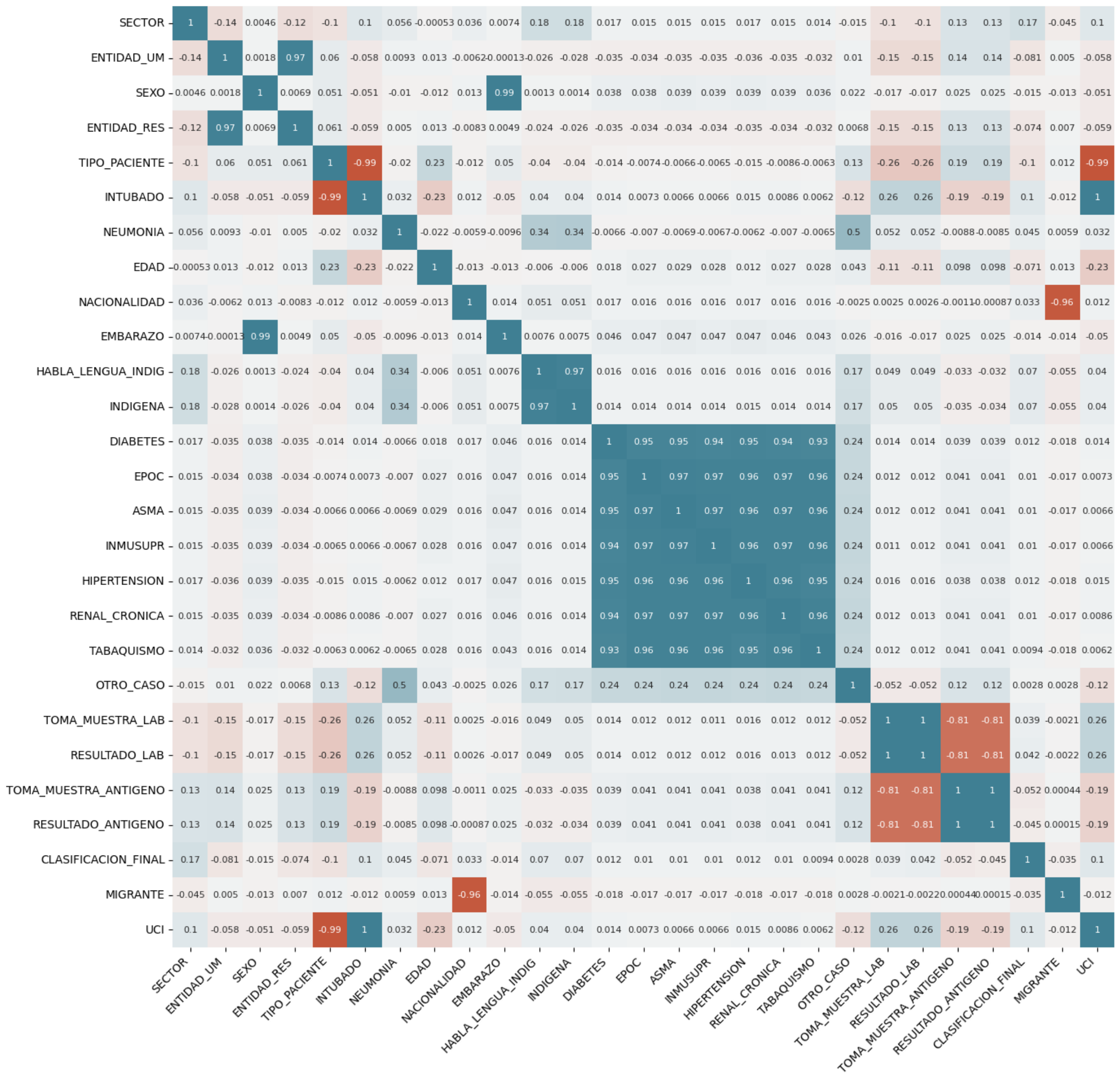


[11]:

```
corr_matrix_2 = numeric_df.corr(method='pearson')

# Print corr matrix as a pretty chart of big size
import matplotlib.pyplot as plt
import seaborn as sns


fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(15, 15))
sns.heatmap(corr_matrix_2,annot=True,cbar=False,annot_kws = {"size":␣
 ↪8},vmin=-1,vmax=1,center=0,
    cmap=sns.diverging_palette(20, 220, n=200), square=True,ax=ax)
ax.set_xticklabels(ax.get_xticklabels(),rotation = 45,horizontalalignment =␣
 ↪'right',)
ax.tick_params(labelsize = 10)
```



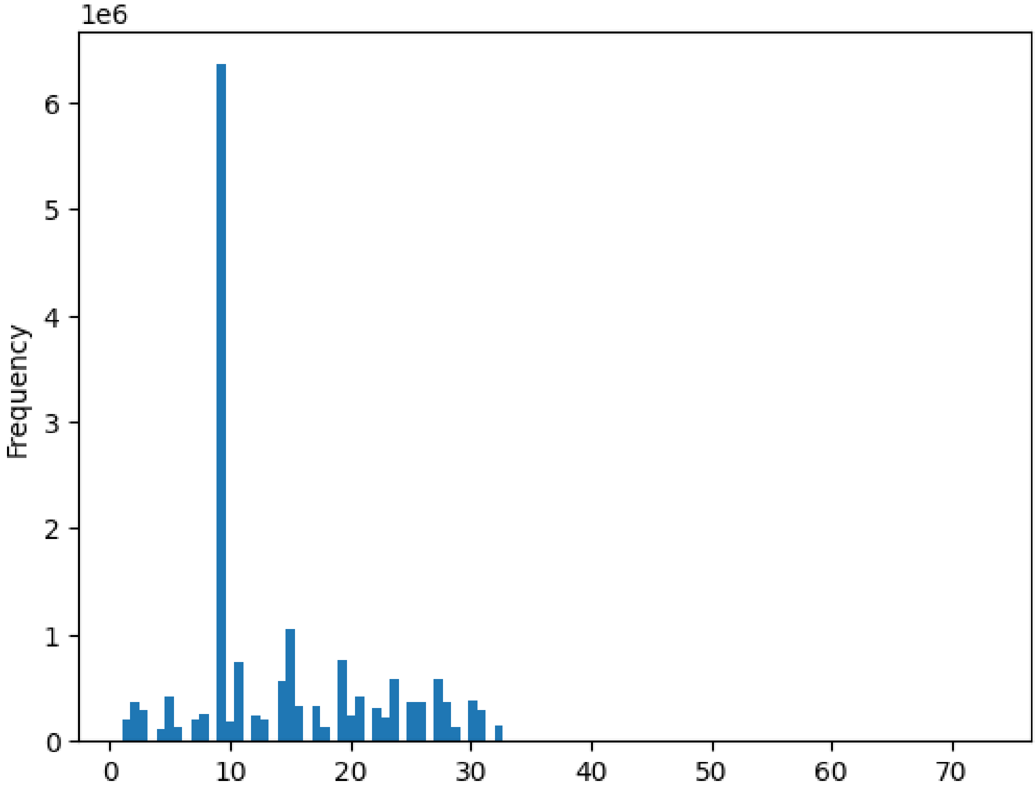
[12]:

```
# Plot frequency distribution of each column in df_numeric

df_numeric['ENTIDAD_UM'].plot.hist(bins=100)
```

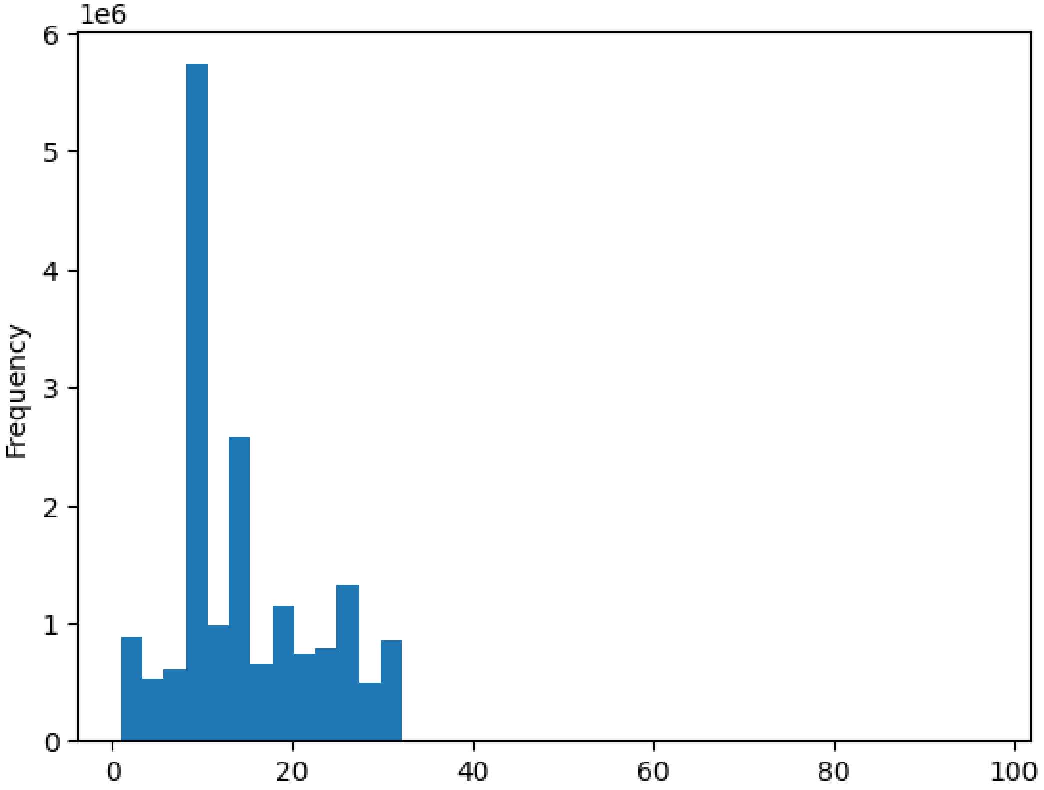
[12]:

```
<Axes: ylabel='Frequency'>
```



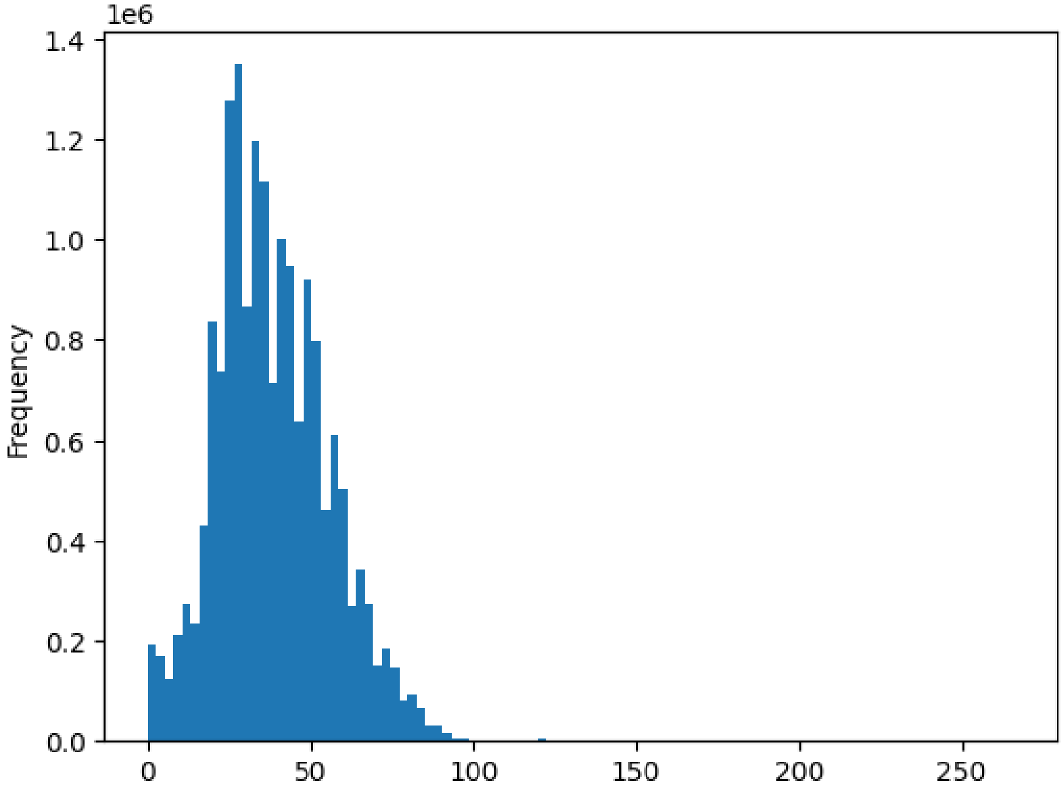
```
[13]: df_numeric['ENTIDAD_RES'].plot.hist(bins=40)
```

```
[13]: <Axes: ylabel='Frequency'>
```



```
[14]: df_numeric['EDAD'].plot.hist(bins=100)
```

```
[14]: <Axes: ylabel='Frequency'>
```



[15]:

```
numeric_df.columns
```

[15]:

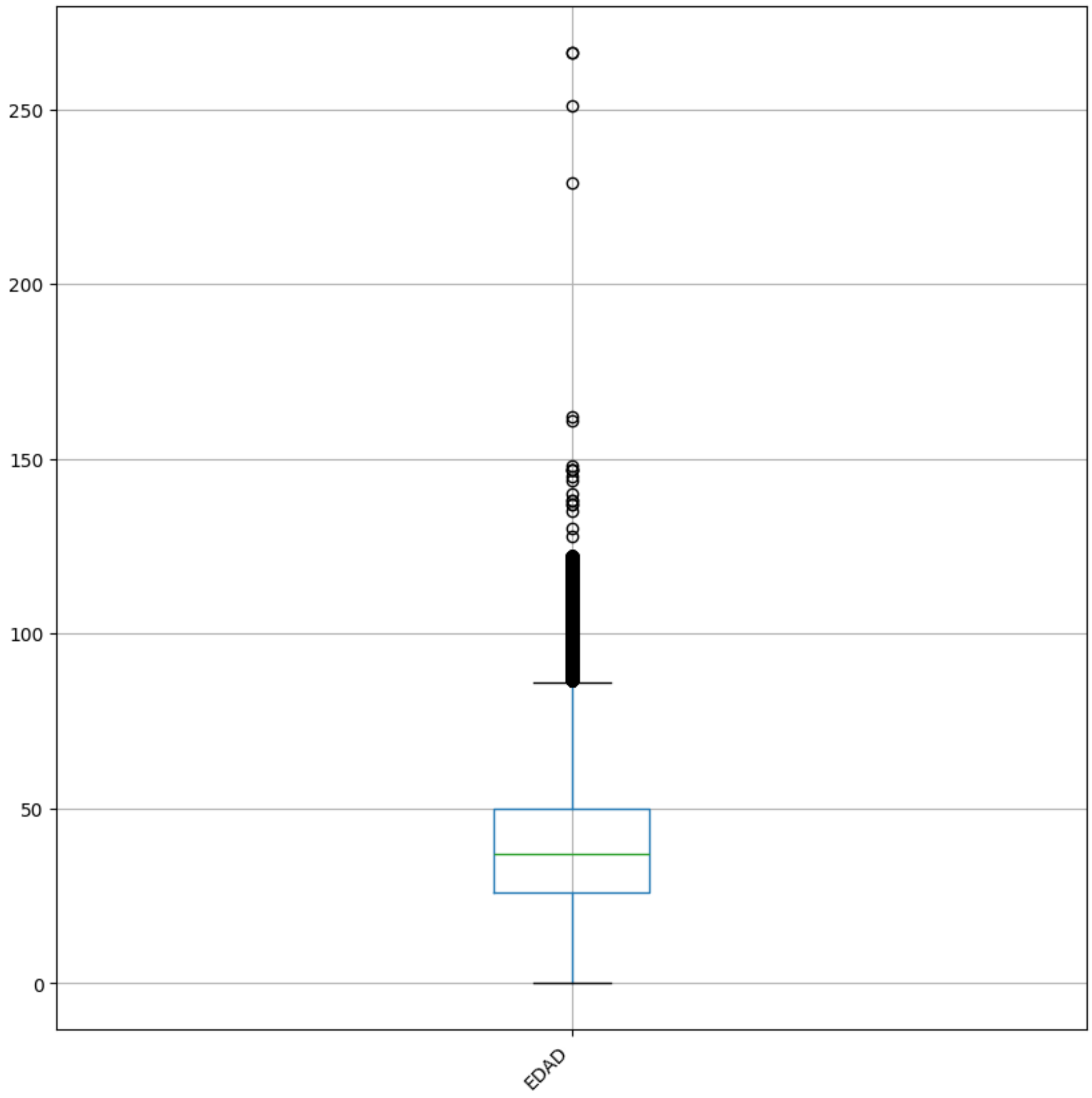
```
Index(['SECTOR', 'ENTIDAD_UM', 'SEXO', 'ENTIDAD_RES', 'TIPO_PACIENTE',
       'INTUBADO', 'NEUMONIA', 'EDAD', 'NACIONALIDAD', 'EMBARAZO',
       'HABLA_LENGUA_INDIG', 'INDIGENA', 'DIABETES', 'EPOC', 'ASMA',
       'INMUSUPR', 'HIPERTENSION', 'RENAL_CRONICA', 'TABAQUISMO', 'OTRO_CASO',
       'TOMA_MUESTRA_LAB', 'RESULTADO_LAB', 'TOMA_MUESTRA_ANTIGENO',
       'RESULTADO_ANTIGENO', 'CLASIFICACION_FINAL', 'MIGRANTE', 'UCI'],
      dtype='object')
```

[21]:

```
# Boxplot of first 10 numeric columns
boxplot_df = numeric_df[['EDAD']]



fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(10, 10))
boxplot_df.boxplot(ax=ax)
ax.set_xticklabels(ax.get_xticklabels(),rotation = 45,horizontalalignment =␣
 ↪'right',)
ax.tick_params(labelsize = 10)
```



2 CATEGORICAL DATA ANALYSIS

[22]:

```
categorical_columns = covid_df.select_dtypes(include=['object']).columns
categorical_df = covid_df[categorical_columns]
categorical_df.describe()
```

[22]:

```
       FECHA_ACTUALIZACION ID_REGISTRO    ORIGEN  ENTIDAD_NAC  MUNICIPIO_RES  \
count             17311026    17311026  17311026     17311024       17311024
unique                   6    17310517         7           68            755
top             2022-07-20     g055540         2            9              7
freq              17311021           2  13246756      5360642        1156760
```

```
       FECHA_INGRESO FECHA_SINTOMAS   FECHA_DEF  OTRA_COM  CARDIOVASCULAR  \
count       17311023       17311022    17311022  17311020        17311020
unique           935            934         934         9               9
top       2022-01-12     2022-01-10  9999-99-99         2               2
freq          120850         132719    16887577  16796892        17036772

        OBESIDAD PAIS_NACIONALIDAD PAIS_ORIGEN
count   17311020          17311015    17311015
unique         8               202         146
top            2            MÃ©xico          97
freq    15737879          17186451    16982716
```

[23]:

```
categorical_df.head()
```

[23]:

```
  FECHA_ACTUALIZACION ID_REGISTRO ORIGEN ENTIDAD_NAC MUNICIPIO_RES  \
0          2022-07-20      z3bf80      2           8            37
1          2022-07-20      z1e370      1          14            85
2          2022-07-20      zze974      1          24            35
3          2022-07-20      zz7067      1           9             7
4          2022-07-20      z1da1e      1           1             1

  FECHA_INGRESO FECHA_SINTOMAS   FECHA_DEF OTRA_COM CARDIOVASCULAR OBESIDAD  \
0    2020-07-28     2020-07-20  9999-99-99        2              2        2
1    2020-04-22     2020-04-18  9999-99-99        2              2        2
2    2021-02-28     2021-02-20  9999-99-99        2              2        2
3    2020-08-18     2020-08-17  9999-99-99        2              2        2
4    2020-03-09     2020-03-05  9999-99-99        2              2        2

  PAIS_NACIONALIDAD PAIS_ORIGEN
0            MÃ©xico          97
1            MÃ©xico          97
2            MÃ©xico          97
3            MÃ©xico          97
4            MÃ©xico          97
```

[37]:

```
entidades_df = pd.read_excel("./assets/Datos COVID/METADATOS/201128 Catalogos.
↪xlsx", engine="openpyxl")
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
File ~/Documents/Master/PatterRecognition/.venv/lib/python3.12/site-packages/
↪pandas/compat/_optional.py:135, in import_optional_dependency(name, extra,␣
↪errors, min_version)
    134 try:
--> 135     module = importlib.import_module(name)
    136 except ImportError:
```

```
File /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/importlib/
 ↪__init__.py:90, in import_module(name, package)
     89         level += 1
---> 90 return _bootstrap._gcd_import(name[level:], package, level)

File <frozen importlib._bootstrap>:1387, in _gcd_import(name, package, level)

File <frozen importlib._bootstrap>:1360, in _find_and_load(name, import_)

File <frozen importlib._bootstrap>:1324, in _find_and_load_unlocked(name,␣
 ↪import_)

ModuleNotFoundError: No module named 'openpyxl'

During handling of the above exception, another exception occurred:

ImportError                               Traceback (most recent call last)
Cell In[37], line 1
----> 1 entidades_df =␣
 ↪pd.read_excel("./assets/Datos COVID/METADATOS/201128 Catalogos.xlsx", engine="openpyxl")

File ~/Documents/Master/PatterRecognition/.venv/lib/python3.12/site-packages/
 ↪pandas/io/excel/_base.py:495, in read_excel(io, sheet_name, header, names,␣
 ↪index_col, usecols, dtype, engine, converters, true_values, false_values,␣
 ↪skiprows, nrows, na_values, keep_default_na, na_filter, verbose, parse_dates,␣
 ↪date_parser, date_format, thousands, decimal, comment, skipfooter,␣
 ↪storage_options, dtype_backend, engine_kwargs)
    493 if not isinstance(io, ExcelFile):
    494     should_close = True
--> 495     io = ExcelFile(
    496         io,
    497         storage_options=storage_options,
    498         engine=engine,
    499         engine_kwargs=engine_kwargs,
    500     )
    501 elif engine and engine != io.engine:
    502     raise ValueError(
    503         "Engine should not be specified when passing "
    504         "an ExcelFile - ExcelFile already has the engine set"
    505     )

File ~/Documents/Master/PatterRecognition/.venv/lib/python3.12/site-packages/
 ↪pandas/io/excel/_base.py:1567, in ExcelFile.__init__(self, path_or_buffer,␣
 ↪engine, storage_options, engine_kwargs)
   1564 self.engine = engine
   1565 self.storage_options = storage_options
-> 1567 self._reader = self._engines[engine](
   1568     self._io,
   1569     storage_options=storage_options,
   1570     engine_kwargs=engine_kwargs,
   1571 )

File ~/Documents/Master/PatterRecognition/.venv/lib/python3.12/site-packages/
 ↪pandas/io/excel/_openpyxl.py:552, in OpenpyxlReader.__init__(self,␣
 ↪filepath_or_buffer, storage_options, engine_kwargs)
    534 @doc(storage_options=_shared_docs["storage_options"])
    535 def __init__(
    536     self,
   (…)
    539     engine_kwargs: dict | None = None,
    540 ) -> None:
    541     """
    542     Reader using openpyxl engine.
    543 
   (…)
    550         Arbitrary keyword arguments passed to excel engine.
    551     """
--> 552     import_optional_dependency("openpyxl")
    553     super().__init__(
    554         filepath_or_buffer,
    555         storage_options=storage_options,
    556         engine_kwargs=engine_kwargs,
    557     )

File ~/Documents/Master/PatterRecognition/.venv/lib/python3.12/site-packages/
 ↪pandas/compat/_optional.py:138, in import_optional_dependency(name, extra,␣
 ↪errors, min_version)
    136 except ImportError:
    137     if errors == "raise":
--> 138         raise ImportError(msg)
    139     return None
    141 # Handle submodules: if we have submodule, grab parent module from sys.
 ↪modules

ImportError: Missing optional dependency 'openpyxl'.  Use pip or conda to␣
 ↪install openpyxl.
```