

# Practica 3.

---

Reporte de análisis Biblioteca ctg\_viz.

---

Análisis y visualización de Datos de Cardiografía.

---

Repositorio:

[https://github.com/AAO-dev/practica3\\_Arellano\\_Oritz\\_Andre](https://github.com/AAO-dev/practica3_Arellano_Oritz_Andre)

Arellano Ortiz Andre.

# 1. Descripción de funciones.

La biblioteca ctg\_viz proporciona un conjunto completo de funciones para el análisis y visualización de datos de cardiotocografía (CTG). A continuación, se describen los módulos principales y sus funciones.

## 1.1 Modulo ctg\_viz.utils

Este módulo contiene funciones de utilidad para la carga y análisis inicial de datos.

**load\_data(filepath:str) → pd.DataFrame**

Carga el dataset desde un archivo CSV.

Parámetros: filepath (str) - Ruta al archivo CSV

Retorna: DataFrame con los datos cargados

**check\_data\_completeness(df: pd.DataFrame) → pd.DataFrame**

Analiza la completitud y calidad de los datos. Calcula conteos de valores nulos, porcentaje de completitud, tipos de datos, valores únicos y estadísticas básicas (desviación estándar, varianza) para columnas numéricas. Clasifica columnas como 'Continua', 'Discreta' o 'Categórica'.

Parámetros: df (pd.DataFrame) - DataFrame de entrada

Retorna: DataFrame resumen con métricas de calidad de datos

## 1.2 Modulo ctg\_viz.preprocessing

Módulo especializado en la limpieza y procesamiento de datos médicos.

**process\_data(df\_input: pd.DataFrame) → pd.DataFrame**

Procesa los datos aplicando tres transformaciones principales:

**1. Eliminación de columnas:** Elimina columnas con más del 20% de valores nulos

**2. Imputación de valores:** Usa la mediana para variables numéricas y la moda para categóricas

**3. Tratamiento de outliers:** Aplica el método IQR (Rango Intercuartílico) con clipping

Parámetros: df\_input (pd.DataFrame) - DataFrame de entrada

Retorna: DataFrame procesado y limpio

## 1.3 Modulo ctg\_viz.plots

Módulo de visualizaciones interactivas utilizando Altair y Plotly. Incluye 9 tipos diferentes de gráficos especializados para datos CTG.

Función	Descripción	Biblioteca
plot_histogram_lb_by_nsp()	Histograma de distribución por clase NSP	Altair
plot_violin_with_swarm()	Gráfico de violín con puntos	Plotly
plot_line_lb_evolution()	Serie temporal de evolución	Altair
plot_scatter_lb_astv()	Diagrama de dispersión LB vs ASTV	Plotly
plot_correlation_heatmap()	Mapa de calor de correlaciones	Altair
plot_bar_class_counts()	Gráfico de barras de conteos	Plotly
plot_boxplot_astv_by_nsp()	Diagrama de caja ASTV por NSP	Plotly
plot_boxplot_uc_by_nsp()	Diagrama de caja UC por NSP	Altair
plot_density_width_by_nsp()	Gráfico de densidad por NSP	Altair

## 2. Visualizaciones Generadas

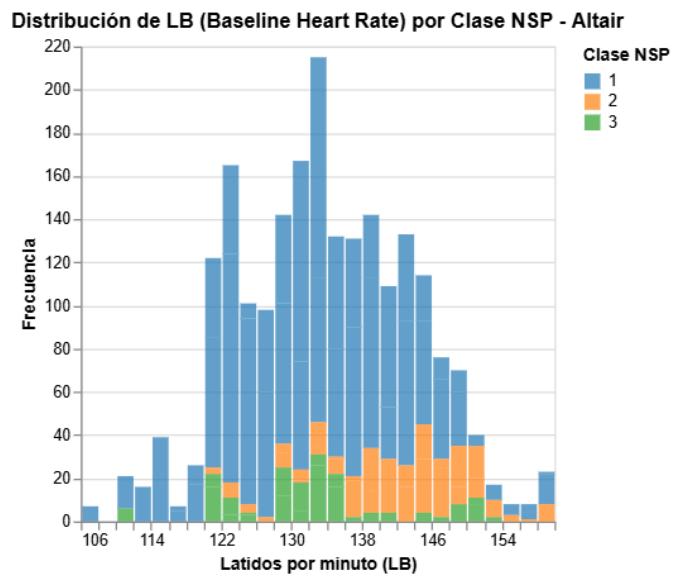
---

**NOTA IMPORTANTE:** para las visualizaciones graficas interactivas puede hacerse tanto en el archivo demostracion.ipynb como en app.py que se explica en el punto 3.6 de este PDF como en el README.md del repositorio.

La biblioteca ctg\_viz genera visualizaciones interactivas de alta calidad que permiten explorar los datos de cardiotocografía desde múltiples perspectivas. A continuación, se describen los principales tipos de visualizaciones disponibles.

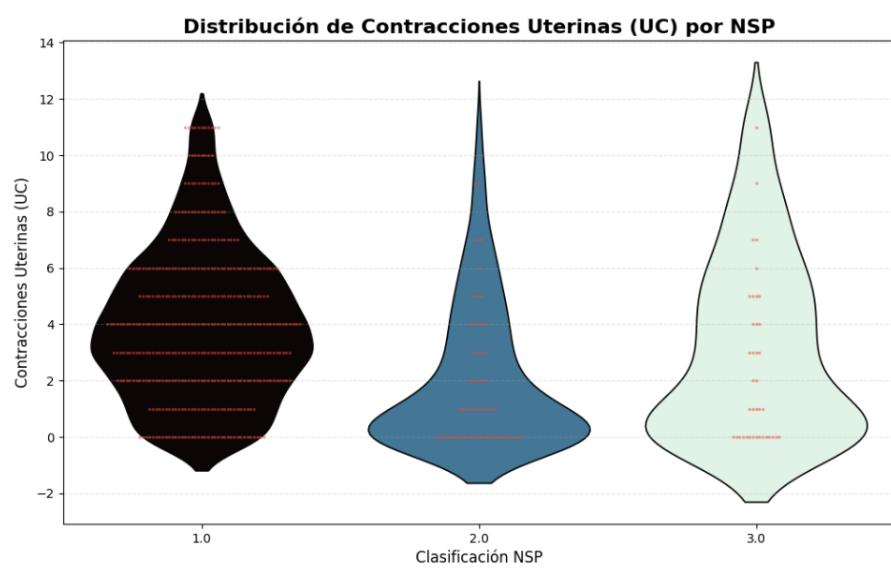
### 2.1 Histograma de LB por NSP

- Función: plot\_histogram\_lb\_by\_nsp
- Librería: Altair
- Representación: Muestra la distribución de frecuencias de la Frecuencia Cardíaca Basal (LB), agrupada por la clase NSP (Normal, Sospechoso, Patológico).
- Utilidad: Permite observar cómo se distribuyen los valores de LB en cada categoría y detectar si existen rangos específicos asociados a condiciones patológicas.



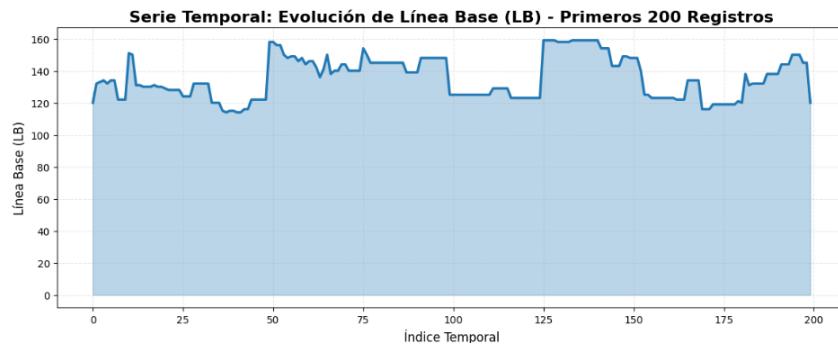
## 2.2 Violin Plot con Swarm Overlay de UC por NSP

- Función: `plot_violin_with_swarm`
- Librería: Seaborn, Matplotlib
- Representación: Combina un gráfico de violín (densidad) con un swarm plot (puntos individuales) de las Contracciones Uterinas (UC) agrupadas por NSP.
- Utilidad: Visualiza la distribución completa de las contracciones, mostrando tanto la densidad de probabilidad como los datos individuales, lo que ayuda a entender la dispersión y el tamaño de la muestra en cada grupo.



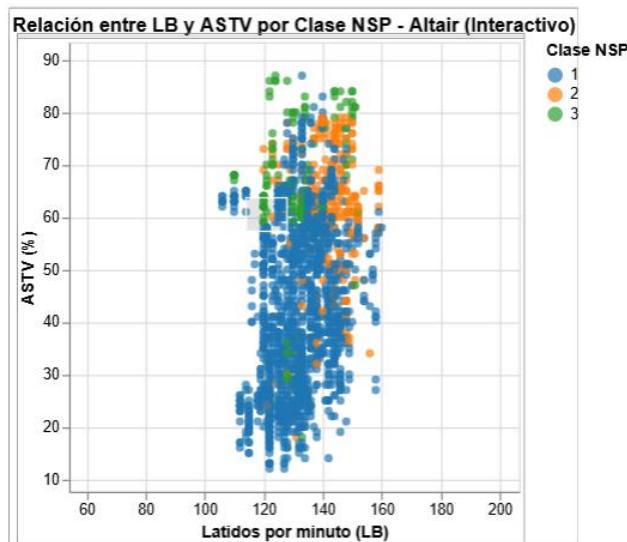
## 2.3 Evolución de LB (Serie de Tiempo Simulada)

- Función: plot\_line\_lb\_evolution
- Librería: Seaborn, Matplotlib
- Representación: Gráfico de línea que muestra la variación de LB a lo largo de un índice secuencial (simulando tiempo).
- Utilidad: Útil para observar tendencias, estabilidad o variabilidad de la frecuencia cardíaca a lo largo de un periodo o secuencia de registros.



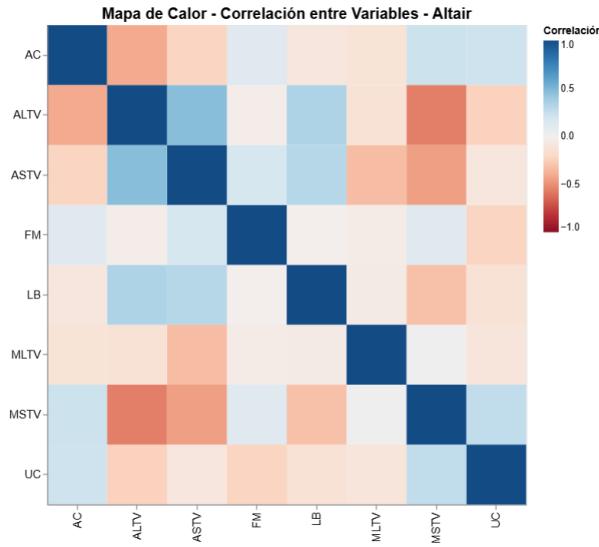
## 2.4 Scatter Plot de LB vs ASTV

- Función: plot\_scatter\_lb\_astv
- Librería: Altair
- Representación: Gráfico de dispersión interactivo que cruza la Frecuencia Cardíaca Basal (LB) con la Variabilidad a Corto Plazo Anormal (ASTV), coloreado por NSP.
- Utilidad: Permite identificar correlaciones o agrupamientos (clusters) entre estas dos variables críticas y ver cómo se separan las clases NSP en este espacio bidimensional.



## 2.5 Mapa de Calor de Correlación

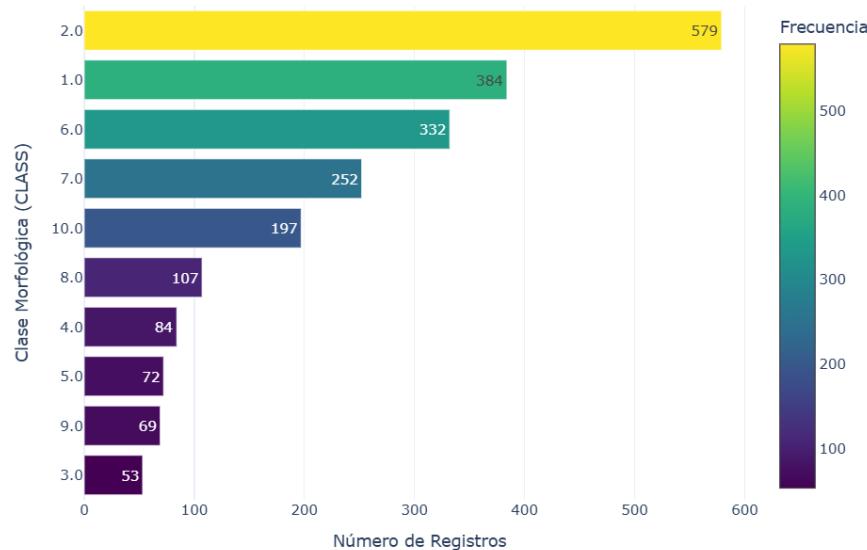
- Función: `plot_correlation_heatmap`
- Librería: Altair
- Representación: Matriz de colores que representa los coeficientes de correlación entre un conjunto de variables seleccionadas.
- Utilidad: Fundamental para identificar relaciones lineales fuertes entre variables, redundancia de información (multicolinealidad) y variables potencialmente predictivas.



## 2.6 Gráfico de Barras de Conteos por Clase

- Función: `plot_bar_class_counts`
- Librería: Plotly
- Representación: Gráfico de barras horizontales que muestra la cantidad de registros para cada Clase Morfológica (CLASS).
- Utilidad: Sirve para analizar el balance de las clases en el dataset, identificando clases mayoritarias y minoritarias.

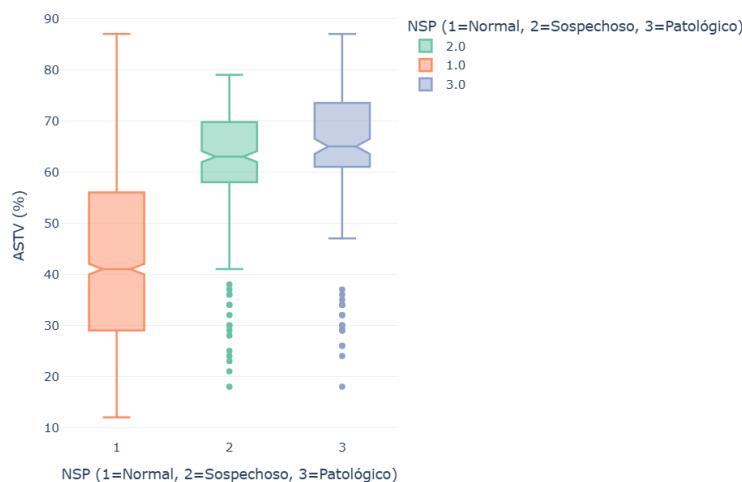
Frecuencia de Registros por Clase Morfológica (CLASS) - Plotly



## 2.7 Boxplot de ASTV por NSP

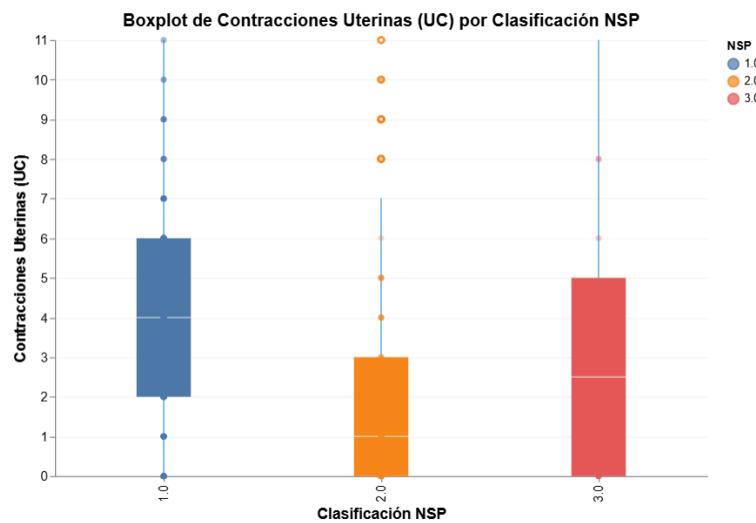
- Función: plot\_boxplot\_astv\_by\_nsp
- Librería: Plotly
- Representación: Diagrama de caja y bigotes de la variable ASTV agrupada por NSP.
- Utilidad: Resume estadísticamente la distribución de ASTV (mediana, cuartiles, valores atípicos) para comparar la variabilidad entre las clases normal, sospechosa y patológica.

Boxplot de ASTV (Abnormal Short Term Variability) por Clase NSP - Plotly



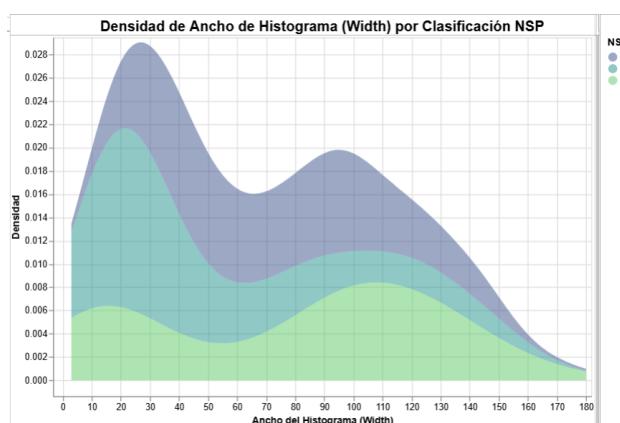
## 2.8 Boxplot UC por NSP

- Función: plot\_boxplot\_uc\_by\_nsp()
- Librería: Altair
- Representación: Muestra un gráfico de cajas combinado con puntos dispersos que representa la distribución de las contracciones uterinas (variable continua UC) agrupadas por la clasificación de salud fetal NSP (variable categórica).
- Utilidad: Comparar distribuciones de contracciones uterinas entre diferentes clasificaciones de salud fetal (NSP).



## 2.9 Gráfico de Densidad de Width por NSP

- Función: plot\_density\_width\_by\_nsp
- Librería: Seaborn, Matplotlib
- Representación: Gráfico de densidad (KDE) de la variable 'Width' (ancho del histograma) para cada clase NSP.
- Utilidad: Permite comparar la forma de las distribuciones de 'Width' entre las clases de una manera más suave y continua que un histograma, facilitando la visualización de solapamientos.



## 3. Procedimientos de Análisis

---

Esta sección describe los procedimientos estándar para el análisis de datos de cardiotocografía utilizando la biblioteca ctg\_viz. Se presentan los pasos metodológicos recomendados para garantizar resultados confiables y reproducibles.

### 3.1 Evaluación Inicial de Calidad de Datos

El primer paso en cualquier análisis de datos CTG consiste en evaluar la calidad y completitud del conjunto de datos mediante la función:

```
def check_data_completeness_Arellano_OritzAndre(df):
```

Este procedimiento genera un informe detallado que incluye:

- Conteo de valores nulos por columna
- Porcentaje de completitud de datos
- Tipos de datos identificados
- Número de valores únicos
- Estadísticas de dispersión (desviación estándar y varianza)
- Clasificación automática de variables (continua, discreta, categórica)

La información obtenida permite identificar columnas con alta proporción de valores faltantes

(>20%) y tomar decisiones informadas sobre su tratamiento posterior.

## 3.2 Preprocesamiento de Datos

Una vez evaluada la calidad de los datos, se procede a aplicar el pipeline de preprocesamiento mediante la función:

```
def process_data(df_input: pd.DataFrame) -> pd.DataFrame:
```

Este procedimiento ejecuta tres transformaciones secuenciales:

### **Paso 1: Eliminación de columnas**

Se eliminan automáticamente las columnas que presentan más del 20% de valores nulos,  
evitando así sesgos derivados de imputación excesiva.

### **Paso 2: Imputación de valores faltantes**

- Variables numéricas: Se imputan con la mediana (método robusto a valores atípicos)
- Variables categóricas: Se imputan con la moda (valor más frecuente)

### **Paso 3: Tratamiento de valores atípicos**

Se aplica el método del Rango Intercuartílico (IQR) con técnica de clipping, que ajusta los valores extremos a los límites calculados sin eliminar observaciones, preservando así el tamaño muestral.

## 3.3 Análisis Exploratorio Visual

El análisis exploratorio se realiza mediante la selección apropiada de visualizaciones según el objetivo del estudio:

### **Para análisis de distribuciones univariadas:**

Utilice histogramas, gráficos de densidad o diagramas de caja para examinar la distribución de variables individuales.

### **Para comparación entre grupos:**

Emplee diagramas de caja o gráficos de violín agrupados por la variable NSP (clasificación clínica) para comparar distribuciones entre categorías.

### **Para análisis de relaciones bivariadas:**

Genere diagramas de dispersión o mapas de calor de correlación para identificar

relaciones entre pares de variables.

**Para análisis temporal:**

Construya series temporales (gráficos de línea) para observar la evolución de variables a lo largo del registro CTG.

**Para análisis de frecuencias:**

Utilice gráficos de barras para visualizar conteos y proporciones de variables categóricas.

### 3.4 Interpretación de Variables Clínicas

La interpretación de resultados debe considerar el significado clínico de las variables CTG:

**LB (Baseline):** Frecuencia cardíaca basal fetal. Rango normal: 110-160 latidos por minuto.

**ASTV (Abnormal Short Term Variability):** Porcentaje de tiempo con variabilidad anormal a corto plazo. Indicador clave del bienestar fetal.

**UC (Uterine Contractions):** Número de contracciones uterinas por segundo. Relacionado con el nivel de estrés fetal.

**NSP (Fetal State Class):** Clasificación clínica del estado fetal en tres categorías: Normal, Sospecha y Patológico.

Los patrones observados en las visualizaciones deben evaluarse en función de estos rangos de referencia clínica y su relevancia médica.

### 3.5 Pruebas Unitarias

Se realizaron las pruebas unitarias mediante pytest en 4 funciones.

```
PS D:\Practicas\practica3_Arellano_OritzAndre> python -m pytest tests/
=====
platform win32 -- Python 3.13.5, pytest-9.0.1, pluggy-1.6.0
rootdir: D:\Practicas\practica3_Arellano_OritzAndre
collected 4 items

tests\test_ctg_viz.py .... [100%]

===== 4 passed in 0.97s =====
```

## 3.6 Utilización de Herramientas Interactivas

Las visualizaciones generadas con Altair y Plotly ofrecen capacidades interactivas que facilitan la exploración detallada de los datos:

- **Zoom y desplazamiento:** Permite examinar regiones específicas de interés
- **Información emergente (tooltips):** Muestra valores exactos al posicionar el cursor
- **Filtrado dinámico:** Facilita el aislamiento de subgrupos de datos
- **Selección de elementos:** Permite identificar observaciones individuales

Estas funcionalidades son particularmente útiles para el análisis de casos específicos y la validación de hipótesis sobre subpoblaciones de interés.

## 3.7 Aplicación Web para Análisis Colaborativo

La biblioteca incluye una aplicación web desarrollada con Streamlit que puede ejecutarse mediante el comando en el bash:

```
python -m streamlit run app.py
```

Esta interfaz proporciona:

- Carga interactiva de conjuntos de datos
- Selección dinámica de variables y parámetros de análisis
- Generación instantánea de visualizaciones
- Capacidad de exportación de resultados