STENOSIS QUANTIFICATION PIPELINE I

ASSUMPTIONS

- **All input information is available and has the correct format.**
- **The Ela’s methodology to obtain the reference values and %DS are consistent and reliable.**
- **The “hyperparamers” given to compute the reference values (Window Size, Range 1, Range 2) are optimal for %DS.**

1.Data Ingestion

- **Input**:
  - **{Px,Py,Pz}:** Point P coordinates.
  - **Dmin:** Diameter of the best fit circle in point P.
  - **Dfit:** Diameter of the inscribing circle in point P.
  - **Area:** Sectional Area in point P.
- **Output**: **Python DataFrame I**

What To Do?
Read input data of centerline properties for an artery or vessel segment points and build a dataframe to store the information.

Later I show how the DataFrame would look like.

2.Reference Value Computation

- **Input**: **Python DataFrame I, Window Size, Range 1 and Range 2**
- **Output**: **Python DataFrame II**

What To Do?
From the previous structured DF, compute both reference values for each point/row:
- Dd: Distal Reference Value
- Dp: Proximal Reference Value

Using 2 different methods from Ela’s work:
- Method 1: Value
- Method 2: Range

This would produce a triplet for each DF point/row defined as: (Dd, Dp, Value) & (Dd, Dp, Range).

Transform the input DataFrame by adding these new computed values for each point/row.

Later I show how the DataFrame would look like.

3. Stenosis %DS Computation

- **Input**: **Python DataFrame II**
- **Output**: **Python DataFrame III**

What To Do?
From thr previous DF information, compute the %DS for every point/row using Ela’s formula, and using each possible value for variable p:

$$DS\% = \left(1 - \frac{p}{\left(\frac{dp + dd}{2}\right)}\right) \times 100$$

p = {**Dmin**, Dfit, Area}

Transform the input DataFrame by adding the p value used and the new computed value for each point/row.

Later I show how the DataFrame would look like.

STENOSIS QUANTIFICATION PIPELINE II

ASSUMPTIONS

- **All input information is available and has the correct format.**
- **The Ela's methodology to obtain the reference values and %DS are consistent and reliable.**
- **The "hyperparamers" given to compute the reference values (Window Size, Range 1, Range 2) are optimal for %DS.**

4.Data Aggregation

- **Input: Python DataFrame I**
- **Output: Python DataFrame IV**

What To Do?
Each DataFrame row represents a point and its behavior in a certain region of an artery.

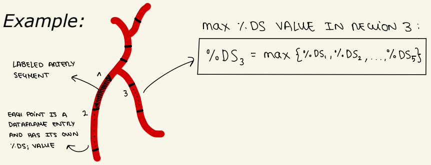
Given vessel labels, points can be splitted by regions. Transform the DF by adding the label region of the artery for each point/row and adding the following new feature that shows the maximum %DS whithin a range of labeled artery points:

for each point/row with same label and artery_id:

max_%DS = max{point(i).%DS, max_%DS}

Since there are several values of %DS for each point depending on (p and method used to compute reference values Dd and Dp), there will also be several values for the previous maximum within each label range.

Example:


LABELED ARTERY SEGMENT
EACH POINT IS A DATAFRAME ENTRY AND HAS ITS OWN %DS; VALUE
1
2
3
MAX %DS VALUE IN REGION 3:
$\%DS_3 = \max\{\%DS_1, \%DS_2, \dots, \%DS_5\}$

This can be useful to later visualize which labeled sectors of the artery have more stenosis risk.

Later I show how the DataFrame would look like.

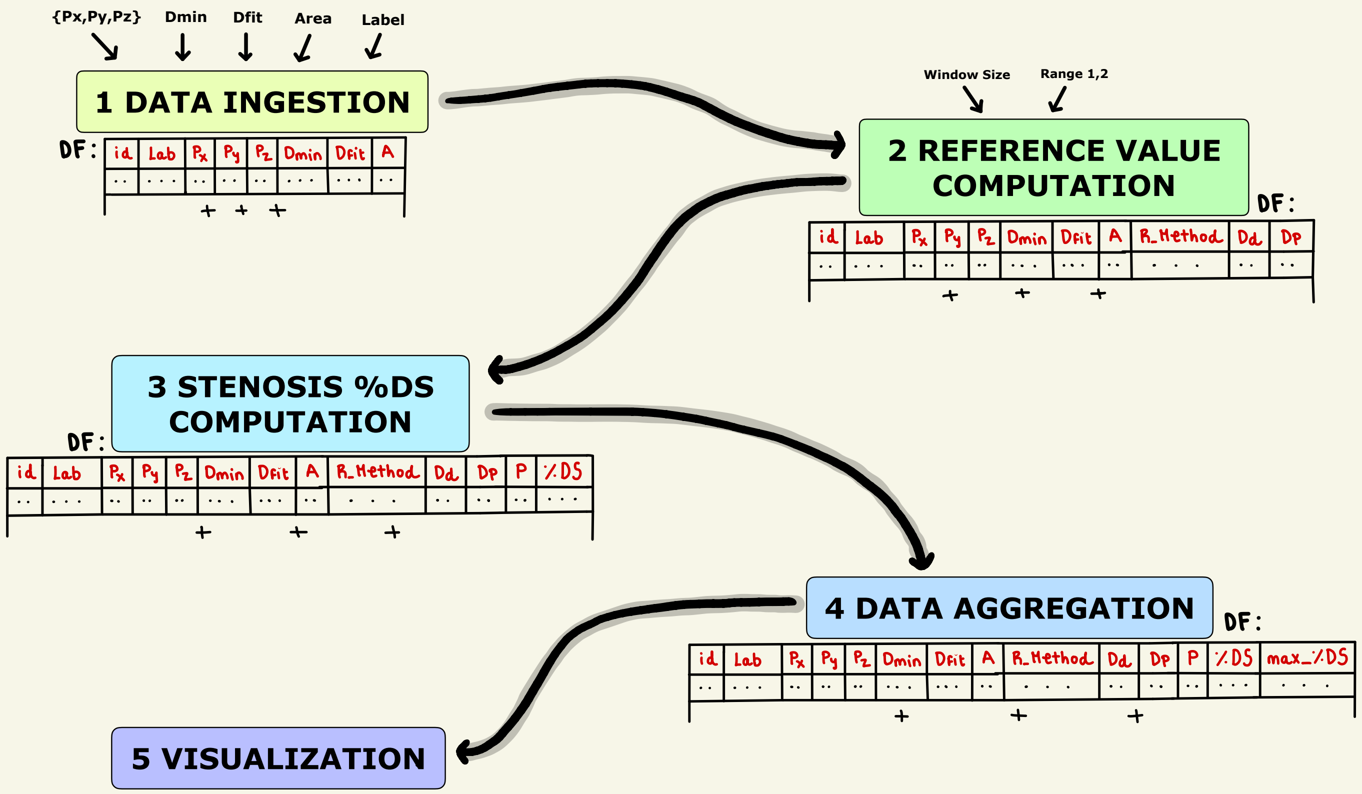
5. Visualization

- **Input: Python DataFrame IV**
- **Output: Python Non-Interactive & Interactive Visualizations**

What To Do?
Use all the initial data of the coronary artery and the computed metrics for each point and regions to create several different visualizations to facilitate the process of analyzing the stenosis at certain region.

I am currently investigating which visual representations could be meaninfgull for the results representation of the given task.

STENOSIS QUANTIFICATION GENERALIZED PIPELINE



DF:

id	Lab	P_x	P_y	P_z	Dmin	Dfit	A
..	...	..	..	..	...	...	..

+ + +

DF:

id	Lab	P_x	P_y	P_z	Dmin	Dfit	A	R_Method	Dd	Dp
..	...	..	..	..	...	...	..	. . .	..	..

+ + +

DF:

id	Lab	P_x	P_y	P_z	Dmin	Dfit	A	R_Method	Dd	Dp	P	%DS
..	...	..	..	..	...	...	..	. . .	..	..	..	...

+ + +

DF:

id	Lab	P_x	P_y	P_z	Dmin	Dfit	A	R_Method	Dd	Dp	P	%DS	max_%DS
..	...	..	..	..	...	...	..	. . .	..	..	..	...	. . .

+ + +