

- EDI ANSI x12 Parsing
- Vendors: FedEx and UPS
 - Specifically for the x12 format 210 and 110 for freight invoices
- Code: written in Python 3.9
 - Libraries: pandas
- Structure of EDI:
 - Each EDI file contains mandatory and optional details Each EDI follows the following structure:
 - Interchange Control Header (ISA)
 - Functional Group Header (GS)
 - Transaction Set Header (ST)
 - Detail Segment
 - Transaction Set Trailer (SE)
 - Functional Group Trailer (GE)
 - Interchange Control Trailer (IEA)
 - A more complex example:
 - Interchange Control Header (ISA)
 - Functional Group Header (GS)
 - Transaction Set Header (ST)
 - Detail Segment
 - Transaction Set Trailer (SE)
 - Transaction Set Header (ST)
 - Detail Segment
 - Transaction Set Trailer (SE)
 - Functional Group Trailer (GE)
 - Functional Group Header (GS)
 - Transaction Set Header (ST)
 - Detail Segment
 - Transaction Set Trailer (SE)
 - Functional Group Trailer (GE)
 - Interchange Control Trailer (IEA)
- Explanation of Structure:
 - Each EDI contains one Interchange Control
 - Within an Interchange Control, multiple Functional Group can exist

- Within a Functional Group, multiple Transaction Sets can exist
- Within a Transaction Set, many Detail Segments can exist
- Separators:
 - Each line is separated by a segment terminator which is located at the 105 index of each EDI file
 - Each set of data is separated by the data separator which is usually “*”
- Structure in Context:
 - For FedEx and UPS:
 - Each EDI file contains only one set of Interchange Control and Functional Group
 - Each Transaction Set contains the information of a single invoice
 - Each Detail Segment contains all the charge information for that invoice
 - Multiple Transaction Sets can exist in a single Functional Group
 - This indicates that multiple invoices would exist in a single EDI file
- Parsing:
 - Approach:
 - In FedEx and UPS EDI, ST has the highest hierarchy since it signifies an invoice
 - This method does not collect data/ charges by invoice, but by delivery number since each delivery number contains its own charges
 - Each invoice can have 1 or more delivery numbers
 - Method:
 - The EDI file is split by the segment terminator
 - Using the resulting list, the program iterates through each line
 - The main key words where data is pulled are:
 - ST
 - This is a reset point
 - Whenever ST appears, the data should reset since ST signifies a new invoice
 - LX
 - This is a reset point
 - Whenever LX appears, the dataset can be created since LX signifies the beginning of a Delivery Number
 - B3

- Information available for extraction:
 - Invoice number
 - At index 2
 - Invoice date
 - At index 6
- N1-BT
 - Information available for extraction:
 - Bill To Address
 - Iterate and concatenate all information from N1 - N4 or until the next keyword is not between N1- N4
- N9 (UPS)
 - Information available for extraction:
 - Account Number
 - At index 2
- N9-CN (UPS)
 - Information available for extraction:
 - Tracking Number
 - At index 2
- N9-CR (UPS)
 - Information available for extraction:
 - Reference Number
 - At index 2
- N9-AW (FedEx)
 - Information available for extraction:
 - Tracking Number
 - At index 2
- L0
 - UPS
 - Information available for extraction:
 - Gross Weight (when 'N' is at index 5)
 - At index 4
 - Actual Weight (when 'B' is at index 5)
 - At index 4
 - FedEx

- Information available for extraction:
 - Gross Weight
 - At index 4
 - Number of Packages
 - At index 8
- L1
 - Information available for extraction:
 - Charge code
 - At index 8
 - Charge description
 - At index 12 if it exists
 - Charge amount
 - At index 4
- N1-SH
 - UPS
 - Information available for extraction:
 - Sender Address
 - Iterate and concatenate all information from N1 - N4 or until the next keyword is not between N1- N4
 - FedEx
 - Information available for extraction:
 - Receiver Address
 - Iterate and concatenate all information from N1 - N4 or until the next keyword is not between N1- N4
- N1-CN (FedEx)
 - Information available for extraction:
 - Sender Address
 - Iterate and concatenate all information from N1 - N4 or until the next keyword is not between N1- N4
- N1-ST (UPS)
 - Information available for extraction:
 - Receiver Address
 - Iterate and concatenate all information from N1 - N4 or until the next keyword is not between N1- N4

- Using these keywords, one can splice the line by the data separator and then pick the info that is needed
- Schema:
 - For Header Dataframe:
 - {
 - "Vendor" : []
 - "Invoice Number" : []
 - "Invoice Date" : []
 - "Account Number" : []
 - "Bill To Address" : []
 - "Line Item" : []
 - "Tracking Number" : []
 - "Reference Number" : []
 - "Gross Weight" : []
 - "Actual Weight" : []
 - "# of Packages" : []
 - "Sender Address" : []
 - "Receiver Address" : []
 - }
 - For Charge Dataframe:
 - {
 - "Invoice Number" : []
 - "Tracking Number" : []
 - "Line Item" : []
 - "Charge Code" : []
 - "Charge Description" : []
 - "Charge Amount" : []
 - }