# Review of Crewai

## Importing data

The recommended methodology is to use CSV\_search\_tool to read in raw claims data

csv\_search\_tool = CSVSearchTool(r'C:\Users\masho\OneDrive\AI\Deloitte\_framework\2\_Analysis\Attempt4\_Crewai\sample\_motor\_claims\_data\_lessdata.csv')  #https://github.com/alexfazio/crewAI-quickstart/blob/main/crewai\_sequential\_CSVSearchTool\_quickstart.ipynb

* Weakness
  + It cant take in 1000 claim entries (It returns “ Inserting batches in chromadb: 0%” then crashes
  + It did work when I decreased the claim entries.
    - Data used: sample\_motor\_claims\_data\_lessdata
  + If one does not use CSV\_search\_tool, the triangle formulation does not work well. Each run produces different results.
    - I used Json format and the values had hallucinations
    - I directly input the data in the tasks prompt which did give results but after fact checking, the results were not adequate due to many errors.
* Recommendation:
  + For version 1 of our tool, lets rather assume that the claims triangles are already available.
  + In later versions we can then work on optimizing the data importing of raw claims data.

## Metadata

### Data and files used during the above analysis

***2\_3\_Crew\_framework\_ChatGPT\_Visio\_Usingtools\_successwithcaveat:***

***Issues***

* The CSVSearchtool couldn't read 1000 claim entries, thus there is a large limitation on the creation of the triangles.
* I tested this by using the 2 excel files, one with 1000 entries, and the other with 13 entries. The CSV with less data read in atleast.

*Summary of what code does:*

* This code integrates data analysis, actuarial science, and automation using the CrewAI framework. The script first imports necessary libraries like pandas, numpy, scipy, matplotlib, and seaborn for data manipulation and visualization, as well as chainladder for actuarial loss reserving calculations. It then sets up several tools (DirectoryReadTool, FileReadTool, SerperDevTool, CSVSearchTool) to handle tasks such as reading data from CSV files, searching directories, and working with external data.
* The core of the script defines multiple CrewAI agents with specific roles, such as "Information Gatherer," "Data Analyst," "Development Factor Specialist," "Risk Adjustment Specialist," and "Report Compiler." Each agent is tasked with performing different aspects of actuarial analysis, from gathering information from the CSV file to performing reserving calculations using the Chain Ladder method. The code constructs tasks for each agent and assigns them to handle data processing in a sequential manner. Finally, the Crew object coordinates the execution of these agents and tasks, producing a comprehensive actuarial analysis report that integrates various calculations, analyses, and risk assessments. The code is structured to handle multiple Lines of Business (LOBs) and efficiently automate the process of reserving and report generation in an actuarial context

***2\_5\_Crew\_framework\_base\_with\_CSVtoinput\_Failed:***

*Issues*

* Tried different ways of inputting the CSV file directly as an input, rather than as part of the prompt in tasks ( where one can pass it as json or dataframe)
* Hallucination risk was high. Most times it gave examples of how one would run it as opposed to using the data I fed in.

What code does:

* This code integrates data processing and analysis for an actuarial project using Python and the CrewAI framework. It begins by defining several functions to handle CSV data in different formats: converting a CSV file into a DataFrame, converting it into a list of dictionaries, and creating a structured list that includes required fields such as 'description' and 'expected\_output.' These functions ensure that the CSV data is correctly formatted for use within the CrewAI framework, making it suitable for further analysis by various agents.
* The main part of the code sets up different agents with specific roles, such as a Data Analyst, Development Factor Specialist, Risk Adjustment Specialist, and Report Compiler, each assigned particular tasks related to actuarial data analysis. The code defines a series of tasks for these agents, such as analyzing the data, creating loss triangles using the Chain Ladder method, calculating risk adjustments using bootstrapping techniques, and compiling a comprehensive actuarial report. The Crew object manages the execution of these tasks sequentially, resulting in a final actuarial analysis output that adheres to regulatory standards like IFRS 17 and APN 401. This process is designed to automate and streamline the end-to-end actuarial data analysis workflow