## Senior Data Engineer Skills Test

### **Task Objective:**

To develop a robust data analysis and visualisation solution using synthetic health data from Synthea. The solution will provide insights into patient demographics, diagnoses, and medical trends, ensuring actionable results for decision-making and reporting.

### **Task Evaluation Criteria:**

This task will be evaluated on the following criteria:

* **Code Quality**: Clear, efficient, and well-documented code. Use of proper version control practices (commit history, branching, descriptive commit messages).
* **Data Wrangling**: Your ability to clean, transform, and manipulate health data accurately and efficiently.
* **Analysis**: Depth of analysis and ability to derive meaningful insights from the data.
* **Visualisation**: Quality and interactivity of the visualisations.
* **Version Control Skills**: Demonstrated proficiency in using Git for version control, creating branches, managing pull requests, and documenting your work on GitHub.
* **Documentation**: Clear README and meaningful commit messages to explain your work.

### **Task Requirements:**

You will need to source data files from the **Synthea** – a synthetic health dataset, which generates synthetic electronic health record (EHR) data. The data is publicly available from the Synthea's official page <https://synthea.mitre.org/downloads>

For simplicity, select the CSV format and files that include the following columns:

* patient\_id
* age
* gender
* diagnosis\_code
* diagnosis\_description
* medication\_code
* medication\_name
* procedure\_code
* procedure\_name
* visit\_date
* hospital\_id

You can work with a sample dataset containing a subset of these columns or a larger dataset as needed.

Please prepare code in R, SQL or Python preferably to answer the following questions.

**Data Wrangling and Cleaning**:

1. Load the dataset into your environment.
2. Inspect the data for missing values, inconsistencies, and data types
3. Ensure data consistency:

* Standardise column names
* Address any inconsistencies in categorical columns
* Check for duplicate records or invalid entries

1. Transform the data as needed, including:

* Creating new variables (e.g., age groups, length of stay etc.)
* Aggregating or pivoting data (e.g., number of visits per patient, frequency of certain diagnoses etc.)

**Exploratory Data Analysis (EDA)**:

1. Conduct an initial exploratory analysis to understand the dataset’s characteristics, such as:

* The distribution of patients by age, gender, or diagnosis\_description.
* The most frequent diagnoses and medications.
* Trends in medical procedures over time.
* Explore the relationship between patient demographics and diagnosis or medication types.

1. Perform basic statistical analysis:

* Calculate mean, median, and standard deviation for numerical columns (e.g., age, frequency of visits).
* Identify any outliers or skewed distributions.

**Visualisation:**

1. Create an interactive report/dashboard with at least three visualisations that will allow a user to explore the synthetic health data. **Be creative!**

* Use interactive elements like dropdowns, filters, or sliders to allow users to explore different aspects of the dataset
* Visualisations should include a mix of chart types (e.g., bar charts, line charts, scatter plots, pie charts, heatmaps etc.)
* Ensure the layout is clear, with intuitive navigation and easily interpretable charts.

**Insight Generation:**

1. Based on your analysis and visualisations, provide key insights into the synthetic health data. For example:

* Are there any age groups or gender categories that are more prone to certain diagnoses?
* Are there any temporal trends in hospital visits or medication use?
* What are the most common diagnose
* s or procedures for patients within the dataset?
* Can you identify any relationships between patient demographics and health outcomes (e.g., diagnoses, treatment success)?

**Code Versioning and Repository Management:**

1. Set up a Git repository on GitHub for version control.

* Create a new public repository
* Clone the repository to your local machine and initialise the project structure

1. Commit your work regularly:

* As you complete each major step (e.g., loading data, cleaning data, creating visualisations), make sure to commit changes with descriptive commit messages.
* Include relevant comments in your code to explain key steps in the data wrangling process, any assumptions made, and key findings.

1. Branching:

* Create separate branches for different tasks (e.g., data-cleaning, exploratory data analysis, visualisation).
* After completing each branch’s task, merge it into the main branch using a pull request (PR), following the GitHub flow.

1. Document the repository:

* Provide a detailed README.md file that explains:
  + The purpose of the repository
  + The dataset used and how it was processed
  + The tools and libraries used
  + Instructions on how to run the analysis and interact with the visualisations
* Optionally, include a Jupyter Notebook or R Markdown file that combines code and markdown cells to explain your approach and findings.
* Keep the repository organised with clear folder structures (e.g., data/, scripts/, notebooks/).

### **Task Deliverables:**

1. **GitHub Repository**:
   * A public GitHub repository with your project files.
   * All code and documentation should be properly version-controlled and organised.
2. **Python or R Script**: The core code files used for data wrangling, analysis, and visualisation.
3. **Jupyter Notebook (Python) or R Markdown (R)**: A file that includes the code, visualisations, and markdown explanations.
4. **README.md**: A detailed description of the project, including how to run the analysis and relevant explanations.

**Notes for Applicants:**

* **GitHub Workflow**: If you are unfamiliar with GitHub, refer to [GitHub’s official documentation](https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests) for help on how to work with repositories, branches, and pull requests.
* **Libraries**: Feel free to use any libraries or packages you are comfortable with in Python (Pandas, NumPy, Matplotlib, Seaborn, Plotly) or R (dplyr, ggplot2, tidyr).
* **Collaboration Best Practices**: Even though this task is individual, the Git version control process simulates a collaborative environment. Ensure you follow best practices, such as breaking tasks into smaller units and committing regularly.

**At the start of your interview, you will be asked to present a 5 minutes (maximum) verbal presentation to the panel on your results from this task.**

Items you may consider including in your presentation include:

* Methods used to explore and understand the data
* Key insights and results from the data
* Demonstration of any interactive functionalities included in your visualisation
* Any data quality issues that may have been discovered

**Please send in your GitHub Repository link, and any other related documentation**