***Analisi dati FHIR in Orange***

Programming for Data Science

Alfonso Marino

Indice

[OBIETTIVI DEL PROGETTO 3](#_Toc157612314)

[INTRODUZIONE AGLI STRUMENTI 4](#_Toc157612315)

[*FHIR* 4](#_Toc157612316)

[*Orange* 6](#_Toc157612317)

[IL PROGETTO 7](#_Toc157612318)

[*Le risorse* 7](#_Toc157612319)

[*Configurazioni preliminari* 8](#_Toc157612320)

[*Definizione di un widget* 9](#_Toc157612321)

[*Widget di selezione dei file* 9](#_Toc157612322)

[*Widget di trasformazione in formato tabellare dei file* 11](#_Toc157612323)

[***Observation Resource*** 11](#_Toc157612324)

[***Descrizione funzionale del widget*** 11](#_Toc157612325)

[I WORKFLOW 17](#_Toc157612326)

[*Analisi descrittiva di Patient Resource* 17](#_Toc157612327)

[*Analisi dei sondaggi PHQ-2 e Tobacco Smocking Status NHIS* 21](#_Toc157612328)

[*Classificazione del BMI* 24](#_Toc157612329)

[CONCLUSIONI 27](#_Toc157612330)

[SITOGRAFIA 29](#_Toc157612331)

# **OBIETTIVI DEL PROGETTO**

Il progetto si propone di esplorare e analizzare le risorse FHIR (*Fast Healthcare Interoperability Resources*) attraverso l’uso combinato del tool *Orange* e di *Python*. La crescente adozione di FHIR nel contesto sanitario moderno ha reso essenziale sviluppare strumenti flessibili e accessibili che facilitino l'analisi di tali informazioni nel quotidiano. Il progetto mira a migliorare l'efficienza delle operazioni quotidiane, consentendo agli eventuali operatori sanitari di accedere in modo agevole a informazioni dettagliate, rilevanti e prontamente comprensibili. Attraverso la realizzazione di widget personalizzati in Python e l'integrazione con Orange, il progetto si propone di semplificare il processo di estrazione dei dati FHIR, garantendo al contempo una visualizzazione chiara e una comprensione approfondita delle informazioni mediche. In questo modo, si punta a incrementare la qualità delle decisioni cliniche e a favorire la collaborazione tra i professionisti della salute, contribuendo così a un ambiente sanitario più efficiente e centrato sul paziente.

Il progetto, pur nascendo con finalità didattiche, presenta il potenziale di diventare un punto di partenza per future evoluzioni. La sua struttura flessibile e la possibilità di implementare ulteriori funzionalità non solo lo rendono adatto all'apprendimento degli utenti, ma anche suscettibile di essere adottato e sviluppato da professionisti del settore sanitario. La sua natura dinamica potrebbe contribuire a colmare lacune analitiche specifiche nel contesto delle risorse FHIR, offrendo un valore aggiunto anche per esigenze professionali più avanzate e promuovendo l'innovazione nel campo dell'informatica sanitaria.

# **INTRODUZIONE AGLI STRUMENTI**

## ***FHIR***

Immagine che contiene testo, schermata, software, Pagina Web

Descrizione generata automaticamenteFast Healthcare Interoperability Resources (FHIR) è uno standard per lo scambio elettronico di informazioni sanitarie. Sviluppato da *Health Level Seven International* (HL7), FHIR è progettato per migliorare l'interoperabilità e semplificare i sistemi sanitari (“*Fast healthcare interoperability resources”,* Wikipedia*)*. È stato sviluppato in risposta alla necessità di metodi più rapidi, più facili e migliori per scambiare la quantità in rapida crescita di dati sanitari. FHIR si basa su standard Internet ampiamente utilizzati da settori al di fuori dell'assistenza sanitaria, rendendo più facile per gli sviluppatori di software supportare le esigenze sanitarie. FHIR utilizza un approccio modulare, strutturando i dati in *risorse* riutilizzabili come pazienti, laboratori, farmaci, sottoforma di JSON, XML, UML o altri formati. Questo taglio standardizzato consente la condivisione e l'analisi, compresa l'integrazione con sistemi non sanitari (Dreyer, 2023). Le risorse FHIR possono essere singoli pacchetti di informazioni che includono metadati, testo o particolari elementi di dati, ma possono anche essere raggruppate in raccolte che creano documenti clinici (Editorial Staff & HealthITAnalytics, 2023).

Figura 1. Panoramica delle risorse

Un altro aspetto importante è rappresentato dalle API, ovvero un insieme di regole e protocolli che consentono lo scambio di dati sanitari tra diverse applicazioni. Queste API utilizzano un protocollo *RESTful*, che sfrutta i metodi standard HTTP, come GET e POST, per recuperare e manipolare solo i dati pertinenti. Ad esempio, un’operazione di lettura può essere eseguita inviando una richiesta GET a un URL specifico, mentre un’operazione di creazione può essere eseguita inviando una richiesta POST con un corpo di richiesta che contiene i dati della nuova risorsa.

Le API FHIR supportano anche l’uso di codici di stato http per indicare il risultato di un’operazione. Ad esempio, un codice di stato 200 indica che un’operazione è stata eseguita con successo mentre un codice di stato 404 indica che la risorsa richiesta non è stata trovata.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteEsistono diverse librerie e strumenti disponibili per aiutare l'implementazione dell'API FHIR. Ad esempio, *HAPI FHIR* è un'implementazione completa dello standard HL7 FHIR per l'interoperabilità sanitaria.

Figura 2. Esempio di applicazione del metodo GET in Python.

Nel campo della data science, FHIR può essere determinante nella trasformazione dell'analisi sanitaria, consentendo lo sviluppo di modelli di intelligenza artificiale e apprendimento automatico per l'assistenza personalizzata e la diagnostica avanzata. È anche utilizzato per reclutare pazienti per studi clinici e raccogliere dati sui risultati, accelerando lo sviluppo di nuovi farmaci e trattamenti. Tuttavia, il tasso di adozione di FHIR nell'analisi rimane basso a causa di sfide quali le limitazioni delle query e le barriere di competenze elevate. Il suo potenziale di sbloccare informazioni per migliorare la qualità e i risultati dell'assistenza rende FHIR un vero e proprio punto di svolta per l'assistenza sanitaria. Man mano che sempre più fornitori adottano gli standard FHIR, l'analisi fornirà maggiori informazioni sull'assistenza ai pazienti e sui risultati su una scala più ampia che mai (Dreyer, 2023).

## ***Orange***

Orange è un toolkit open source per la visualizzazione dei dati, l'apprendimento automatico e il data mining sviluppato dall'Università di Lubiana. È dotato di un front-end di programmazione visuale per l'analisi qualitativa esplorativa dei dati e la visualizzazione interattiva dei dati (*Orange (software)*, Wikipedia).

I componenti di Orange, noti come *widget*, vanno dalla semplice visualizzazione dei dati, alla selezione di sottoinsiemi e alla preelaborazione, fino alla valutazione empirica degli algoritmi di apprendimento e alla modellazione predittiva. Questa programmazione visuale è implementata attraverso un'interfaccia in cui i flussi di lavoro vengono creati collegando i widget.

 Sebbene Orange sia progettato per essere user-friendly per gli utenti non programmatori, offre anche un aspetto interessante per gli sviluppatori. Infatti, gli utenti avanzati e gli sviluppatori possono sfruttare Orange come una libreria Python per la manipolazione dei dati e personalizzare o estendere i widget secondo le proprie esigenze. Questa versatilità fa di Orange uno strumento adatto sia per coloro che cercano un'analisi dati rapida e intuitiva, sia per chi desidera una maggiore flessibilità e controllo attraverso la programmazione Python.

# **IL PROGETTO**

## ***Le risorse***

L'utilizzo sinergico del software Orange e dati FHIR può offrire vantaggi significativi nell'analisi e nella gestione dei dati sanitari. Nel nostro caso specifico l’obiettivo è stato quello di sviluppare delle analitiche partendo dall’analisi di determinate risorse:

1. **Patient (Paziente)**: Rappresenta le informazioni principali di un paziente, come nome, data di nascita, genere, codice identificativo univoco, ecc.
2. **DiagnosticReport (Rapporto diagnostico)**: Contiene i risultati di test, immagini diagnostiche o osservazioni relative a una diagnosi specifica del paziente.
3. **Encounter (Visita)**: Rappresenta un incontro tra un paziente e un operatore sanitario o una struttura sanitaria. Include informazioni sulla ragione dell'incontro, il luogo, la data, e altre informazioni correlate.
4. **Observations (Osservazioni)**: Contiene misurazioni, osservazioni o valutazioni cliniche relative alla salute del paziente, come la pressione sanguigna, il battito cardiaco, i livelli di glucosio nel sangue, ecc.
5. **Medication (Medicinale)**: Rappresenta un medicinale che può essere prescritto o somministrato a un paziente, incluso il nome, la dose, la via di somministrazione, ecc.
6. **Procedure (Procedura)**: Contiene informazioni su procedure o interventi medici eseguiti o pianificati per il paziente, come un intervento chirurgico, una terapia, ecc.
7. **MedicationRequest (Richiesta di medicinale)**: Rappresenta la richiesta di prescrizione di un medicinale per un paziente da parte di un medico o di un operatore sanitario, inclusi dettagli come il medicinale richiesto, la dose, la frequenza, ecc.

Immagine che contiene testo, schermata, menu

Descrizione generata automaticamenteQueste informazioni sono state estratte da file sintetici in formato JSON ottenuti tramite l’utilizzo di *Synthea*, un generatore di dati FHIR Open Source. Ogni file contiene informazioni relative ad un singolo paziente, ciò vuol dire che oltre alla risorsa di tipo Patient saranno presenti tutte quelle risorse associate a quel particolare individuo. Il documento JSON è caratterizzato da un'intensa gerarchia di dati, con l'impiego di dizionari e liste annidate. Questa organizzazione intricata offre una rappresentazione dettagliata e strutturata dei dati, consentendo una gestione efficace di informazioni complesse. Di seguito un esempio di come si presentano i file JSON:

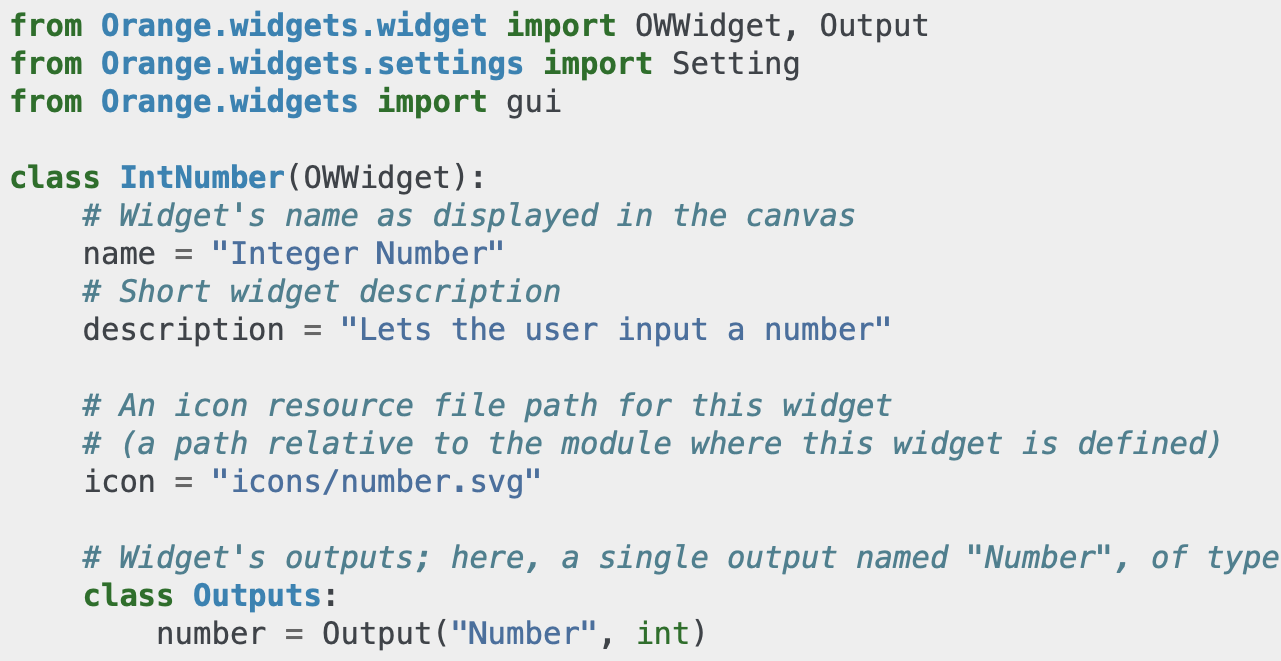
## ***Configurazioni preliminari***

Immagine che contiene testo, schermata, Carattere, bianco

Descrizione generata automaticamenteIl progetto, che chiameremo “*orange-demo*”, è stato strutturato seguendo rigorosi standard per la creazione di un pacchetto Python compatibile con l'ambiente di sviluppo Orange. La seguente organizzazione è stata adottata per garantire una distribuzione agevole e una corretta integrazione dei widget nell'ambiente Orange:

1. **Directory Principale: orange-demo/**
   * Questa directory è il punto principale del progetto e contiene il file *setup.py* insieme a una sottodirectory denominata “*orangedemo”*.
2. **File di Configurazione: setup.py**
   * Lo script *setup.py* è essenziale per la distribuzione e l'installazione del pacchetto. Le informazioni fornite in questo file sono cruciali per definire il pacchetto e le sue dipendenze.
   * Le principali informazioni fornite in *setup.py* includono:
     + **name**: Il nome del pacchetto, nel caso specifico denominato “*Analisi dati FHIR*”.
     + **packages**: indica i pacchetti Python inclusi nel progetto.
     + **package\_data**: definisce i dati aggiuntivi da includere con il pacchetto, come i file SVG nella directory *icons*.
     + **classifiers**: Informazioni aggiuntive sul pacchetto, come il riferimento alla classe di esempio “*Example::Invalid*”.
     + **entry\_points**: La dichiarazione dei punti di ingresso per il pacchetto, indicando che il pacchetto *orangedemo* contiene widget per la categoria “*Analisi dati FHIR*” nell'ambiente Orange.
3. **Sottodirectory del Pacchetto: orangedemo/**
   * Questa directory contiene il modulo *\_\_init\_\_.py*, indicando che è un pacchetto Python, e il modulo, che nella figura è denominato *OWDataSamplerA.py*, ma che rappresenta il file che implementa il widget.

## ***Definizione di un widget***

*OWWidget* è la classe base di un widget in un workflow di Orange. Ogni widget deve includere informazioni come il nome, la descrizione, la categoria e l’icona, nonché definire le capacità di input/output e implementare la sua funzionalità all'interno della tela.

Nell'ambito del design dei widget Orange, è prevista la suddivisione in due zone di lavoro distinte: l'area di controllo e l'area principale. L'area di controllo, posizionata a sinistra, contiene controlli per le impostazioni e le opzioni del widget, mentre l'area principale ospita elementi visivi generati in base agli input e alle opzioni selezionate. Nella classe *OWWidget*, questa suddivisione è resa disponibile attraverso gli attributi “*self.controlArea”* e “*self.mainArea*”. La flessibilità nell'uso di entrambe le aree o solo una di esse consente di personalizzare l'interfaccia utente per rispondere alle specifiche esigenze del widget, mantenendo la chiarezza e la facilità d'uso. Nel nostro caso, abbiamo optato per l'utilizzo esclusivo di *self.controlArea*, mantenendo così una focalizzazione chiara sulla configurazione del widget attraverso i controlli in questa area.

## ***Widget di selezione dei file***

Il primo widget, denominato *OWFhirLoading*, è stato progettato per caricare risorse FHIR da file JSON. L'interfaccia utente è stata costruita sfruttando la libreria *PyQt5.*

Di seguito una descrizione funzionale dettagliata:

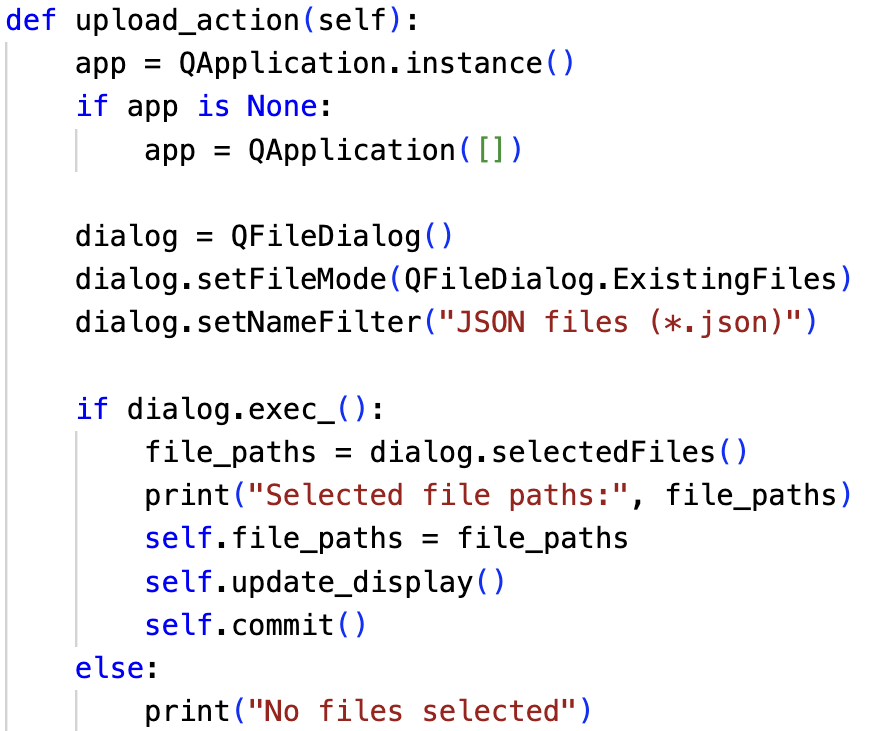
* Immagine che contiene testo, Carattere, schermata

  Descrizione generata automaticamenteIl widget estende la classe *OWWidget* di Orange definendo il nome, la descrizione e l’uscita del widget attraverso la classe *Outputs*, che corrisponderà ad una lista contenente tutti i file JSON selezionati.
* Immagine che contiene testo, Carattere, schermata, numero

  Descrizione generata automaticamenteNel metodo *\_\_init\_\_,* viene inizializzata la lista *file\_paths* per memorizzare i percorsi dei file selezionati e si invoca il metodo *init\_ui ()* per inizializzare l'interfaccia utente.
* Immagine che contiene testo, schermata, Carattere, numero

  Descrizione generata automaticamenteIl metodo *init\_ui ()* crea un layout verticale (*QVBoxLayout*) per la disposizione degli elementi dell'interfaccia utente. Viene aggiunto un pulsante "*Select JSON file(s)*" e un *QTextEdit* per visualizzare i percorsi dei file selezionati. Infine, il layout verticale viene aggiunto all’area di controllo del widget.
* Quando il pulsante viene premuto, viene chiamato il metodo *upload\_action()*, il quale apre una finestra di dialogo (*QFileDialog*) per la selezione di uno o più file JSON.

L'opzione *setFileMode (QFileDialog.ExistingFiles)* consente la selezione di file esistenti multipli. Viene inoltre fornito un filtro per rendere possibile solamente la selezione di file in formato JSON (*setNameFilter ("JSON files (\*. json)")*).

Se l'utente seleziona uno o più file, i percorsi vengono memorizzati nell'attributo *self.file\_paths*, e vengono chiamati i metodi *update\_display ()* e *commit().*

* Immagine che contiene testo, Carattere, schermata, bianco

  Descrizione generata automaticamenteIl metodo *update\_display ()* aggiorna il campo di testo *QTextEdit* con i percorsi dei file selezionati, mentre il metodo *commit()* invia gli output del widget (*list\_of\_paths*) con i percorsi dei file selezionati.

## ***Widget di trasformazione in formato tabellare dei file***

Una volta caricati i file, occorrerà estrarre da questi le risorse di nostro interesse per poi preparare i dati alle successive analisi e alla trasformazione in formato tabellare, rispettando i parametri previsti da Orange per la creazione delle *Data Table*. In questa sede viene fatto un approfondimento in merito alla risorsa *Observation* e al suo widget di caricamento.

### ***Observation Resource***

La risorsa *Observation* è un componente chiave dello standard FHIR, utilizzata per documentare misurazioni, risultati o affermazioni su un paziente, un dispositivo o altri soggetti in ambito sanitario; può essere utilizzata per registrare un'ampia gamma di informazioni cliniche e non cliniche, come i risultati di laboratorio, i segni vitali, i sintomi e persino la storia sociale, come lo stato di fumatore.

La risorsa *Observation*, progettata per catturare misurazioni e valutazioni soggettive tipicamente in un singolo momento temporale, è in grado di gestire valori eccezionali o nulli, inclusi nell'insieme dei valori o mediante l'utilizzo dell'elemento *dataAbsentReason*. Inoltre, le osservazioni possono essere connesse tra loro attraverso elementi come *hasMember* o possono far parte di un'osservazione composita tramite l'elemento *component*.

Gli elementi fondamentali della risorsa, nonché quelli estratti dai file ai fini delle analisi sono:

* **Resource ID**: questo elemento identifica ciò che viene osservato o misurato.
* **Patient ID**: indica il soggetto sul quale vengono effettuate le osservazioni.
* **Encounter ID**: indica il codice che identifica univocamente la visita effettuata.
* **Date**: l'ora o il periodo in cui è stata effettuata l'osservazione.
* **Category**: rappresenta la macroarea di afferenza dell’osservazione, come *Vital Signs*, *Laboratory* e *Survey*.
* **LOINC Description**: descrive l’ambito specifico della categoria di osservazione come *Body Weight*, *Blood Pressure* e *Heart Rate.*
* **LOINC Code**: ad ogni ambito è associato un codice univoco proveniente da un sistema terminologico standardizzato, ovvero [*http://loinc.org*](http://loinc.org)*.*
* **Value**: Il risultato o il valore dell'osservazione.
* **Unit of Measure**: un valore può essere espresso in vari formati, come una quantità, un codice, una stringa, un intervallo, un rapporto, ecc.
* **Component 1** **LOINC Description, Component 2 LOINC Description**: questi campi sono destinati a contenere le descrizioni LOINC delle componenti dell'osservazione. Spesso sono popolati quando l’osservazione riguarda la pressione sanguigna e quindi presenterà le descrizioni per Diastolic Blood Pressure e Systolic Blood Pressure.
* **Component 1 Value, Component 2 Value**: rappresenta il risultato ottenuto dall’osservazione dei componenti.
* **Component 1 Unit of Measure, Component 1 Unit of Measure**: indica l’unità di misura associata ai valori misurati per i componenti dell’osservazione.
* **Status Survey**: corrisponde allo stato del paziente rispetto ad un particolare aspetto proposto dall’osservazione.

### ***Descrizione funzionale del widget***

L’obiettivo finale del widget è quello di ottenere una tabella, con le informazioni della risorsa Observation, che sia visualizzabile nel tool Orange. Questo è stato reso possibile tramite un accurato studio della struttura dei file a disposizione, dal quale ne è conseguita la selezione dei dati necessari, nonché pulizia degli stessi per migliorarne la leggibilità all’utente.

* Immagine che contiene testo, Carattere, schermata

  Descrizione generata automaticamenteAnche in questo caso il widget estende la classe *OWWidget* di Orange definendo il nome, la descrizione, l’entrata che corrisponde alla lista di percorsi derivanti dal widget di caricamento e l’uscita del widget che sarà la tabella contenente le informazioni elaborate dai file JSON.
* Immagine che contiene testo, schermata, Carattere

  Descrizione generata automaticamenteNel metodo *\_\_init\_\_*, vengono inizializzate le liste *string\_variables, cat\_variables, e numeric\_variables* per definire quali variabili e colonne del JSON verranno estratte. Vengono inoltre create le liste *all\_res e all\_keys* per memorizzare le risorse estratte e le chiavi di tutti i dizionari. In questa fase viene sviluppata anche l’interfaccia utente, la quale prevede la creazione di un box all’interno dell’area di controllo, al cui interno vi sarà un campo per l’inserimento dell’URL del server FHIR. Di conseguenza è creato anche un pulsante che, quando premuto, chiamerà la funzione *validate\_api*. Infine, viene creato una label per visualizzare eventuali messaggi di errore.
* Immagine che contiene testo, Carattere, schermata

  Descrizione generata automaticamenteLa funzione *validate\_api* utilizza una *regex* per verificare se l’URL inserito è nel formato di un endpoint FHIR, quindi se è valido chiama la funzione *make\_reques*t, altrimenti mostra un messaggio di errore attraverso *self.display\_message*. La regex verifica se l'input dell'utente corrisponde a un URL che inizia con "http" o "https", seguito da un dominio principale e, opzionalmente, sottodomini e un percorso.
* *Immagine che contiene testo, Carattere, schermata

  Descrizione generata automaticamente*La funzione *make\_request* esegue una richiesta GET all’URL del server FHIR specificato, gestendo eventuali errori durante l’estrazione. Questo viene fatto attraverso l’utilizzo della libreria *requests.* In caso positivo, l viene trasformata in JSON per far si che possa essere chiamata la funzione *extract\_ObservationReques*t per estrarre la risorsa e *flatten\_dict* per appiattire i dati JSON. Quindi viene creata la tabella.
* *Immagine che contiene testo, schermata, Carattere

  Descrizione generata automaticamente*La funzione *extract\_ObservationRequest* accetta un percorso del file JSON o i dati JSON direttamente. Estrae informazioni da un bundle FHIR verificando il tipo di risorsa, se il JSON non è un bundle imposta il flag *add\_prefix* a *True* in modo tale da aggiungere un prefisso ai nomi delle colonne.
* La funzione *flatten\_dict* è progettata per appiattire una struttura di dizionari nidificati in una struttura a chiave/valore piatta. Questo processo di appiattimento è utile quando si lavora con dati JSON complessi, in cui i valori possono essere strutture nidificate di dizionari e liste. La funzione accetta un dizionario *d* e due argomenti opzionali: *key* (che rappresenta la chiave corrente durante la ricorsione) e *sep* (il separatore utilizzato nei nomi delle colonne). La funzione itera attraverso gli elementi del dizionario *d* e, se trova un valore che è a sua volta un dizionario o una lista, chiama ricorsivamente sé stessa per accedere nella struttura annidata. Durante questo processo, i nomi delle chiavi vengono concatenati utilizzando il separatore *sep.* La funzione restituisce un nuovo dizionario appiattito contenente tutte le chiavi e i valori della struttura originale.

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

* Immagine che contiene testo, Carattere, schermata

  Descrizione generata automaticamenteLa funzione *make\_domain* è responsabile della creazione del dominio per la tabella Orange. Utilizza la libreria pandas per creare un DataFrame (*self.df*) dai dati raccolti nelle richieste di osservazione. Successivamente, la funzione costruisce un dominio Orange che definisce i tipi di variabili e i loro nomi in base alle liste *self.string\_variables, self.numeric\_variables, self.cat\_variables*. La struttura del dominio è organizzata in variabili stringa, numeriche e categoriche.
* Immagine che contiene testo, Carattere, schermata

  Descrizione generata automaticamenteLa funzione *make\_cat\_variables* è specifica per la creazione delle variabili categoriche del dominio. Estrae le colonne categoriche rilevanti dal DataFrame *self.df* e crea variabili discrete Orange da esse. Restituisce una lista di variabili categoriche da aggiungere al dominio.
* Immagine che contiene testo, Carattere, schermata

  Descrizione generata automaticamenteLa funzione *create\_table* svolge diverse operazioni. Utilizza il dominio creato con *make\_domain* per definire la struttura della tabella a partire dai dati appiattiti nel DataFrame. Successivamente si crea una lista di lista, con l’ausilio della libreria *numpy,* contenente le righe della tabella Orange. Ogni valore delle liste è convertito in stringa tramite la map. In aggiunta a ciò, la funzione filtra le righe della tabella in base a una condizione specifica, ovvero la presenza del codice LOINC “93025-5”. Successivamente, modifica i nomi delle colonne attraverso *modify\_column\_names* e i valori della tabella tramite *modify\_table\_values*. Infine, invia la tabella elaborata attraverso l'uscita *processed\_table*.
* La funzione *modify\_table\_values* prende una tabella Orange come input e rimuove il prefisso “*urn:uuid*:” dai valori nelle colonne “*resource\_subject\_reference*” e “*resource\_encounter\_reference”*, invocando la funzione *remove\_uuid\_prefix*.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteLa funzione *remove\_uuid\_prefix* accetta un valore come input e, se il valore contiene un prefisso “*urn:uuid*:”, lo rimuove restituendo il valore senza il prefisso. Viene utilizzata per pulire i valori nelle colonne dalla rappresentazione UUID, per facilitare il merge tra tabelle di risorse differenti.

* Immagine che contiene testo, schermata, Carattere

  Descrizione generata automaticamenteLa funzione *modify\_column\_names* modifica i nomi delle colonne della tabella Orange utilizzando un mapping predefinito (*column\_name\_mapping*). La mappatura fornisce nuovi nomi significativi alle colonne per una migliore comprensione del significato dei dati.
* Immagine che contiene testo, Carattere, schermata

  Descrizione generata automaticamenteIl decorator *@Inputs.list\_of\_paths* è una caratteristica specifica di Orange, che può essere utilizzata in Python per modificare il comportamento di un'altra funzione o metodo. In questo caso, viene utilizzato per specificare che la funzione *set\_input* del widget accetta un input denominato *list\_of\_paths*, il quale è associato ai percorsi dei file JSON che vengono selezionati nel widget di caricamento e forniti come input al widget in questione. Quando l'utente fornisce un valore all'input *list\_of\_paths*, il decoratore assicura che la funzione *set\_input* venga chiamata con questo valore. In particolare, l'input viene passato come argomento alla funzione *set\_input*. La funzione *set\_input* estrae i dati dalle richieste di osservazione nei file JSON specificati, li appiattisce e li elabora attraverso le funzioni descritte sopra per creare e inviare la tabella elaborata attraverso l'uscita *processed\_table.*

**I WORKFLOW**

I workflow che sono stati sviluppati prendono in esame un campione di cento file in formato JSON contenenti ognuno risorse che descrivono la situazione clinica di un paziente.

***Analisi descrittiva di Patient Resource***

Immagine che contiene diagramma, linea, testo, schermata

Descrizione generata automaticamente

Il primo workflow è stato incentrato sull’analisi descrittiva della risorsa Patient con grafici e tabelle.

Dopo aver caricato le risorse utili alla disamina, tramite i widget precedentemente descritti, viene utilizzato il widget *Data Info* per comprendere la struttura della tabella generata. In questa prima fase, la tabella si compone per 100 righe e 91 colonne, tra cui 5 di carattere numerico e 86 di tipo stringa. Pertanto, è stato necessario l’utilizzo del widget *Select Columns* per lavorare solamente con le informazioni di nostro interesse come, ad esempio, il nome e il cognome del paziente, data di nascita o lo stato civile. Successivamente con il widget *Edit Domain*, si è provveduto a modificare i nomi delle colonne, per facilitarne la comprensione, e a modellare appropriatamente le categorie delle variabili. Una volta eseguite queste operazioni di pulizia del dataset, sono state realizzate alcune rappresentazioni utili a descrivere i dati presenti al suo interno. Una prima categorizzazione può essere fatta in base al sesso del paziente; infatti, vengono rilevati 53 pazienti di genere maschile e 47 di genere femminile. In ognuna di queste categorie vi è una maggiore rappresentanza della razza bianca, presente per l’88% nelle donne e nell’85% negli uomini, a discapito di razze come quella nera o Afro Americana, Nativi hawaiani o di altre isole del Pacifico e Asiatica.

Nell’analisi del genere dei pazienti, si è evidenziato anche l’eventualità che gli individui siano sposati o meno. Dal grafico sottostante si evince come le percentuali per ogni tipologia di stato civile siano pressoché simili per genere; infatti, si ha una percentuale di circa 55% individui sposati per entrambi i sessi.

Immagine che contiene Rettangolo, schermata, diagramma, Diagramma

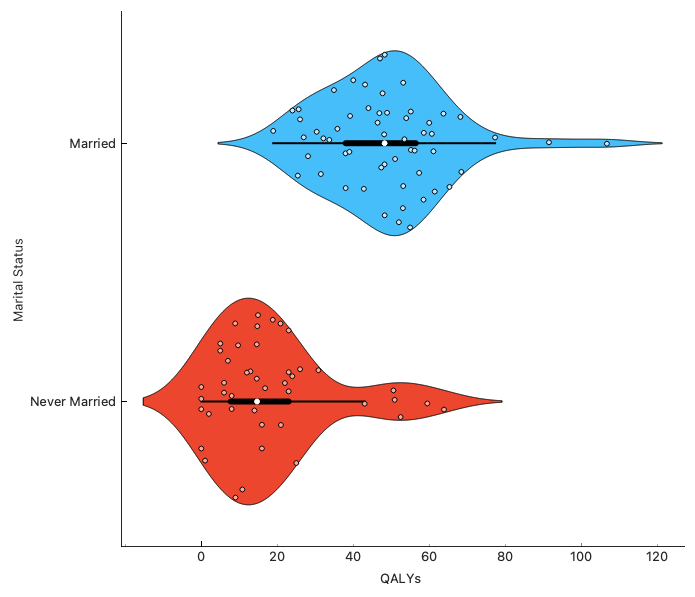
Descrizione generata automaticamenteImmagine che contiene schermata, Rettangolo, giallo

Descrizione generata automaticamente

Distribuzione del genere per razza

Distribuzione di probabilità dello stato civile dato un genere

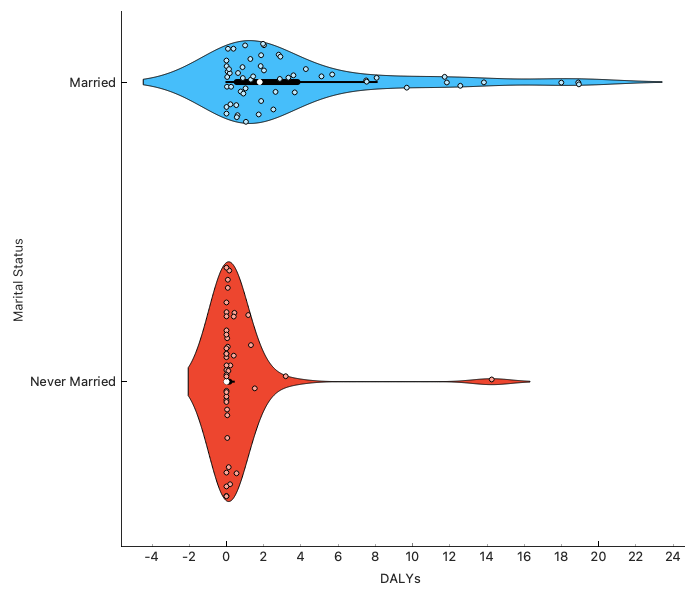
Altri aspetti interessanti analizzati sono gli indici QALYs (*Quality-Adjusted Life Years*) e DALYs (*Disability-Adjusted Life Years*). Per QALYs si intende un valore che misura la quantità di anni di vita aggiustati in base alla qualità, attribuendo un valore inferiore a quegli anni vissuti con una qualità di vita ridotta a causa di malattie o disabilità; mentre DALYs fa riferimento alla somma di anni vissuti con disabilità e anni di vita persi a causa di morte prematura. Tramite il widget *Group by* si è calcolata la media di questi due valori in funzione del genere e dello stato civile ed è subito evidente come i valori per le persone sposate sono sensibilmente più alti.

Immagine che contiene testo, schermata, Carattere, numero

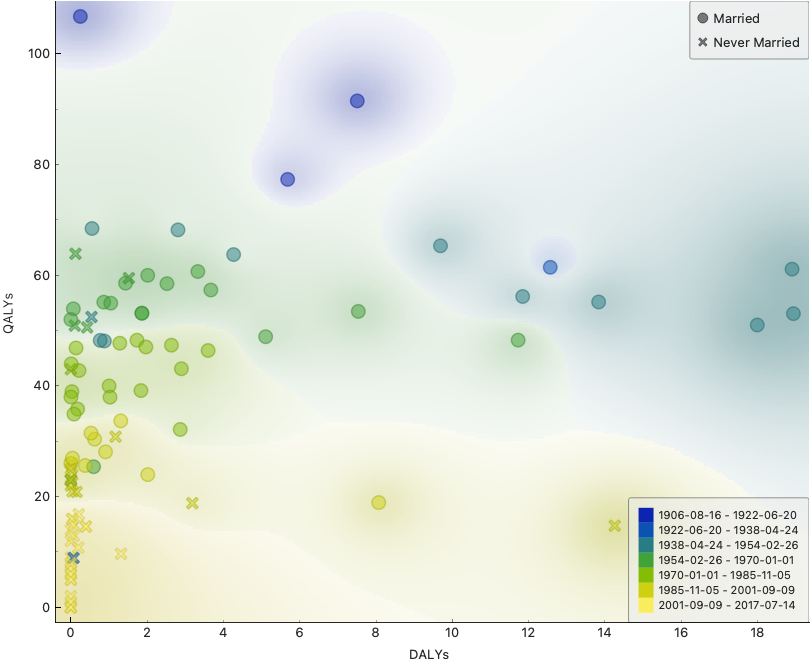
Descrizione generata automaticamente

Violin plot di QALYs per Marital Stats

Group by per genere e stato civile per il calcolo delle medie di QALYs e DALYs



Violin Plot di DALYs per Marital Status

La ragione di queste differenze nella distribuzione dei valori è da attribuire al fatto che, nel campione, alle persone con stato civile *Never Married*, sono associate spesso anche le date di nascita più recenti, e quindi di conseguenza i valori tendono maggiormente verso lo 0 dati i pochi anni di vita dalla registrazione del record.

Violin Plot di QALYs per Marital Status

Scatterplot tra DALYs e QALYs in funzione dello stato civile e della data di nascita

## ***Analisi dei sondaggi PHQ-2 e Tobacco Smocking Status NHIS***

Immagine che contiene diagramma, schermata

Descrizione generata automaticamenteIl workflow numero 2 riguarda la disamina relativa alle informazioni derivanti dai sondaggi effettuati tra i pazienti, in particolare in merito al PHQ-2 e Tobacco Smocking Status NHIS.

L’intento di questo flusso di lavoro è quello di spiegare come i risultati dei sondaggi sono distribuiti tra pazienti di genere diverso.

I sondaggi PHQ-2 e il Tobacco Smoking Status NHIS sono strumenti utilizzati in ambito medico e sanitario per raccogliere informazioni specifiche relative alla salute mentale e all'abitudine al fumo:

* Il *PHQ-2* è uno strumento di screening utilizzato in campo medico per individuare sintomi depressivi in un individuo. Esso consiste in due domande brevi, alle quali bisogna attribuire un valore da 0 a 3, la somma di questi valori può indicare la presenza di sintomi depressivi, suggerendo la necessità di ulteriori valutazioni o interventi da parte di professionisti della salute mentale (Kroenke et al., 2003).
* Il sondaggio sullo *stato del fumo di tabacco del NHIS* è parte di un più ampio National Health Interview Survey e si focalizza sulla raccolta di informazioni riguardo all'abitudine al fumo di tabacco degli individui. Tali informazioni sono cruciali per monitorare i tassi di fumo nella popolazione e per sviluppare strategie preventive al fine di ridurre l'uso del tabacco e migliorare la salute pubblica (Jha et al., 2013)

Ai fini dell’analisi, sono state importate sia le informazioni relative alla risorsa Patient che Observation, sulle quali sono state effettuate operazioni di pulizia e di gestione dei dati. Nel caso di Patient, sono stati presi in considerazione l’ID del paziente, nome e cognome, data di nascita, genere, stato civile e razza, e successivamente trattati con il widget *Edit Domain* per configurare le tipologie delle variabili e adottare un nome più consono ad ognuno.

Per quanto riguarda Observation, una volta caricati i file ed estratta la risorsa di riferimento, sono state selezionate le colonne relative all’ID del paziente, alla categoria dell’osservazione, al codice LOINC e la sua descrizione letterale, il valore misurato e la sua unità di misura e al risultato di un eventuale sondaggio.

Anteprima della tabella di Patient Resource

Immagine che contiene testo, diagramma, schermata

Descrizione generata automaticamenteQueste tabelle sono state unite per mezzo del widget *Merge Data*, nello specifico le colonne della tabella Patient sono state aggiunte a quella di Observation in base all’ID del paziente. Orange mette a disposizione anche il widget *Feature Statistics*, per avere una panoramica generale delle statistiche di base relative alle variabili.

Output del widget Feature Staistics

Una volta definito l’ordine di visualizzazione delle colonne, in accordo con le finalità inizialmente indicate, vengono selezionate solamente quelle osservazioni inerenti alla categoria *survey.* La tabella ottenuta è sottoposta ad una *Pivot Table*, tramite la quale si evince come i sondaggi PHQ-2 e il Tobacco Smoking Status NHIS sono quelli con più occorrenze nel campione, rispettivamente 498 e 1002.

Per quanto riguarda l’analisi del PHQ-2, sono state selezionate le righe per cui fosse presente il sondaggio ed è stata rimossa la colonna relativa a Status Survey in quanto non presentava valori. Per ogni paziente il sondaggio è stato ripetuto più volte, pertanto, tramite il widget *Group By*, le osservazioni sono state raggruppate per paziente in modo da ottenere un valore medio per ognuno. Il risultato ottenuto è ben visualizzabile tramite un Violin Plot, dal quale si evince come, sebbene la media tra i generi sia molto simile, le donne fanno registrare una dispersione di valori più ampia, il che potrebbe rappresentare una base per futuri approfondimenti per ricercarne le ragioni.

Immagine che contiene diagramma, origami, creatività

Descrizione generata automaticamente

Distribuzione dei valori medi del PHQ-2 per genere

Immagine che contiene testo, schermata, Carattere, ricevuta

Descrizione generata automaticamentePer l’analisi del sondaggio inerente al fumo, sono state selezionate le relative righe, nonché sono state prese in considerazioni le osservazioni di pazienti nati prima del 2006, per evitare che soggetti in età infantile potessero influenzare i risultati. Come scritto sopra, per ogni paziente è presente più di un’osservazione; quindi, per facilitare l’analisi è stato utilizzato il widget *Unique* in modo da evitare la ripetizione dei pazienti, tenendo conto della prima istanza. I risultati del sondaggio sono sintetizzabili tramite una *Pivot Table*, che evidenza come la maggior parte dei pazienti intervistati, indipendentemente dal sesso, non abbiano mai fumato.

## ***Classificazione del BMI***

Immagine che contiene diagramma

Descrizione generata automaticamente

Il terzo workflow è stato sviluppato con l’obiettivo di classificare gli individui in base al loro BMI (Body Mass Index) e definire qual è la categoria con più occorrenze nel campione.

Il *BMI* è un dato biometrico utilizzato come indicatore dello stato di peso di un individuo. È calcolato come il rapporto tra il peso di una persona (in chilogrammi) e il quadrato della sua altezza (in metri) (*Gindro*, 2021).

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamenteA tal fine, è stata sufficiente il caricamento della sola risorsa Observation, dalla quale sono state estratte le colonne relative all’ID del paziente, alla categoria dell’osservazione, il codice LOINC e la sua descrizione, nonché il valore misurato, in modo tale da poter prendere in considerazione solamente quelle righe in cui occorreva il valore “*Body Mass Index*”. Successivamente, si è provveduto ad ottenere un valore medio di BMI per ogni paziente, tramite il widget *Group By*.

Anteprima output Group By

Il BMI è utilizzato per classificare le persone in diverse categorie di peso, secondo i criteri definiti dall'*Organizzazione Mondiale della Sanità* (OMS):

* Grave magrezza: < 16,00
* Sottopeso: 16,01 - 17,50
* Leggermente sottopeso: 17,51 - 18,50
* Normopeso: 18,51 - 25,00
* Sovrappeso: 25,01 - 30,00
* Obesità di I classe (lieve): 30,01 - 35,00
* Obesità di II classe (moderata): 35,01 - 40,00
* Obesità di III classe (grave): > 40,00

Immagine che contiene schermata, testo, software, Pagina Web

Descrizione generata automaticamentePer conciliare questa classificazione con i dati a nostra disposizione, è stato utilizzato il widget *Feature Constructor*, il quale consente di aggiungere manualmente colonne al set di dati, partendo dai valori già esistenti. Consente di scegliere il tipo della caratteristica (discreta, continua o stringa) e i suoi parametri (nome, valore, espressione). Per le variabili continue è richiesta la costruzione di un’espressione in Python tramite l’uso del costrutto *if-else*. Nel caso specifico, la colonna che è stata creata ha rispettato i parametri previsti da tale funzione:

Finestra di dialogo del widget Feature Constructor

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamenteUna volta fatto ciò, la tabella generata assumerà questa forma:

Anteprima della tabella con la nuova colonna

Tramite una *Pivot Table* otteniamo quello che è il conteggio delle occorrenze per ogni categoria, ben rappresentato tramite un istogramma. Ciò che si evince è che la maggior parte delle persone comprese nel campione hanno una indice di massa corporea più alto del dovuto; in future analisi si potrebbero individuare possibili conseguenze di questo stato oppure ancora capire se vi sia una correlazione con il territorio in cui queste persone vivono.

Distribuzione delle categorie di BMI

Immagine che contiene Diagramma, schermata, linea, diagramma

Descrizione generata automaticamente

# **CONCLUSIONI**

La presente relazione ha esplorato l’uso innovativo del framework Orange in combinazione con le risorse FHIR, focalizzandosi sull’implementazione di widget personalizzati in Python e descrivendone possibili utilizzi pratici attraverso tre workflow. L’obiettivo principale è stato sviluppare strumenti analitici avanzati che possano adattarsi alle specifiche esigenze dell’analisi dei dati sanitari.

L'approfondita descrizione dei widget personalizzati ha evidenziato la loro flessibilità e adattabilità per rispondere alle sfide analitiche complesse presentate dai dati sanitari. Questi widget hanno dimostrato di fornire soluzioni specializzate, facilitando l'analisi, la manipolazione e la visualizzazione dei dati in modo efficiente e intuitivo.

Successivamente, sono stati presentati tre workflow che illustrano l'integrazione sinergica di Orange e delle risorse FHIR, evidenziando l'efficacia dell'approccio proposto, coprendo le diverse fasi dell'analisi dei dati sanitari e facilitando una comprensione completa e approfondita delle informazioni cliniche.

In conclusione, questo progetto fornisce una solida base per ulteriori ricerche nel migliorare l'efficienza dell'analisi dei dati sanitari, ottimizzando i processi di decisione clinica e supportando la ricerca scientifica. Le potenzialità emergenti includono l'applicazione di tecniche avanzate di machine learning, la creazione di modelli predittivi più sofisticati e l'esplorazione di approcci innovativi per l'interpretazione dei dati.

# ***SITOGRAFIA***

Wikipedia contributors. (2023, December 24). *Fast healthcare interoperability resources*. Wikipedia. <https://it.wikipedia.org/wiki/Fast_Healthcare_Interoperability_Resources>

Dreyer, B. (2023, December 11). *The future of healthcare analytics relies on FHIR data | Qrvey*. Qrvey. <https://qrvey.com/blog/the-future-of-healthcare-analytics-relies-on-fhir-data/>

Editorial Staff & HealthITAnalytics. (2023, December 27). Breaking down the fast Healthcare Interoperability Resource (FHIR). *HealthITAnalytics*. <https://healthitanalytics.com/news/4-basics-to-know-about-the-role-of-fhir-in-interoperability>

*Use of FHIR in clinical research: From electronic medical records to analysis*. (n.d.). CDISC. <https://www.cdisc.org/kb/articles/use-fhir-clinical-research-electronic-medical-records-analysis>

Wikipedia contributors. (2023, March 28). *Orange (software)*. Wikipedia. <https://it.wikipedia.org/wiki/Orange_(software)>

Health Level Seven International. (s.d.). HL7 Fast Healthcare Interoperability Resources (FHIR). <https://www.hl7.org/fhir/>

Kroenke, K., Spitzer, R. L., & Williams, J. B. W. (2003). The Patient Health Questionnaire-2. *Medical Care*, *41*(11), 1284–1292. <https://doi.org/10.1097/01.mlr.0000093487.78664.3c>

Jha, P., Ramasundarahettige, C., Landsman, V., Rostron, B. L., Thun, M. J., Anderson, R. N., McAfee, T., & Pető, R. (2013). 21st-Century Hazards of Smoking and Benefits of Cessation in the United States. *The New England Journal of Medicine*, *368*(4), 341–350. <https://doi.org/10.1056/nejmsa1211128>

Gindro, R. (2021, February 19). *BMI (Indice massa corporea) e calcolo online*. Valori Normali. <https://www.valorinormali.com/altri-esami/bmi-indice-massa-corporea/>