
# POLITECNICO DI TORINO

*Corso di Laurea Magistrale in Ingegneria Matematica*

*Tesi di Laurea Magistrale*

## Cruscottistica direzionale per il controllo dei laboratori in un contesto sanitario

Candidato
Irene Fassi

Relatrice
Prof.ssa Silvia Anna Chiusano

Marzo 2018

# Indice

# Ringraziamenti

Prima fra tutti desidero ringraziare la mia famiglia, Amabile, Spartaco e Chiara per essermi sempre stati vicini e avermi supportato durante tutto il percorso universitario.

Ringrazio l'azienda Engineering Ingegneria Informatica per avermi permesso di svolgere lo stage aziendale su cui si basa questo lavoro di tesi. Un ringraziamento particolare alla Dott.ssa Isabella Iennaco, che è stata la mia tutor aziendale e che mi ha insegnato molto, e alla Dott.ssa Grazia Cazzin, che mi ha seguita durante la stesura della tesi. Desidero inoltre ringraziare tutti i colleghi dell'ufficio per il supporto e la formazione aziendale.

Ringrazio la Prof.ssa Silvia Anna Chiusano per la disponibilità durante questi mesi di lavoro e per i suggerimenti che mi ha dato.

Desidero ancora ringraziare Domenico che mi ha supportato e sopportato in questi ultimi mesi del mio percorso di studio.

Infine un grazie a tutti i miei colleghi universitari, in particolare a Enrico e Leonardo, con cui ho condiviso questi anni universitari.

# Introduzione

Questo lavoro tratta una piccola parte di quello che è il mondo della Business Intelligenge e, in particolare, si focalizza su un caso di studio in ambito sanitario. Vengono inizialmente esposti i concetti fondamentali della Business Intelligence: dalla raccolta dati alla loro analisi, passando attraverso la spiegazione di come si costruisce e si popola un data warehouse e di quali sono le principali tecniche per estrarre dai dati delle informazioni fruibili da ogni tipo di utente. È importante capire quanto questa disciplina possa portare a grandi benefici per aziende di ogni tipo. La nostra epoca si sta evolvendo sempre più verso il digitale ed è necessario che le aziende imparino a tenere il passo con le nuove tecnologie e i nuovi bisogni. Viene in seguito esposta la teoria statistica che è stata utilizzata per sviluppare il lavoro, concentrandosi quindi sui test di ipotesi e su una tecnica di classificazione chiamata regressione logistica.

Dopo aver introdotto l'azienda Engineering Ingegneria Informatica SpA che ha permesso la creazione di questa tesi e il software BI Knowage usato per la sua elaborazione, si passa all'ultimo capitolo, che rappresenta il cuore di questo lavoro. Viene presentato il progetto Pagoda che gestisce i dati accolti dall'area di Ingegneria Clinica dell'AUSL di Modena e che è l'oggetto dell'indagine effettuata. Una prima parte di essa consiste nella creazione di cruscotti tramite Knowage, al fine di dare una rappresentazione visiva a una piccola parte della mole di dati del progetto. La seconda parte è un'analisi più avanzata fatta tramite tecniche statistiche utilizzando il software R. L'obiettivo di questa analisi è soddisfare alcune delle esigenze esposte dal cliente

del progetto Pagoda:

1. verificare se le soglie di riferimento delle analisi di laboratorio sono corrette in relazione all'evoluzione e al cambiamento della popolazione nel corso del tempo;

2. verificare se l'essere esente o meno da ticket sia correlato alla patologicità del risultato.

# Capitolo 1

# La Business Intelligence

La maggior parte delle aziende possiede enormi basi di dati che contengono dati di tipo operativo; queste basi di dati costituiscono una potenziale miniera di informazioni utili. I sistemi per il supporto alle decisioni consentono di analizzare lo stato dell'azienda per poi intervenire su di esso prendendo migliori decisioni e in modo più rapido. I sistemi per il supporto alle decisioni aziendali servono a diversi scopi: analizzare e predire l'evoluzione della domanda, individuare le aree critiche, avere chiarezza dei conti e trasparenza finanziaria, definire e realizzare strategie vincenti (meno costi e più profitti). La prima definizione di Business Intelligence viene proposta da Howard Dresner, analista di Gartner Grouup, nel 1989, per descrivere gli strumenti informatici in grado di soddisfare le esigenze dei manager aziendali. La definizione generale data da Dresner è la seguente: "Business Intelligence describes the enterprise's ability to access and explore information, often contained in a Data Warehouse, and to analyze that information to develop insights and understanding, which leads to improved and informed decision making. BI tools includes: ad hoc query, report writing, decision support systems (DDSs), executive information systems (EISs) and, often, techniques such as statistical analysis and on line analytical processing (OLAP)".[1]

La Business Intelligence è una disciplina di supporto alle decisioni strate-

[1] Fonte: www.gartner.com

giche aziendali. L'obiettivo di questa è la trasformazione dei dati aziendali in informazioni facilmente fruibili da parte degli attori aziendali a diversi livelli di dettaglio e per applicazioni di analisi. Per fare questo è necessaria un'adeguata infrastruttura hardware e software di supporto. Gli ambiti applicativi della Business Intelligence sono innumerevoli: industrie manifatturiere, servizi finanziari, telecomunicazioni, sanità, etc.

L'evoluzione della Business Intelligence è continua e cresce sia dal punto di vista qualitativo, sviluppando soluzioni sempre più avanzate per soddisfare i bisogni dell'impresa, sia dal punto di vista quantitativo. Si possono identificare diverse fasi di sviluppo qualitativo della Business Intelligence. La prima fra tutte è sicuramente un livello di reporting che ha lo scopo di dare una visione generale e unificata delle operazioni aziendali e dei loro risultati. Si sta poi facendo sempre più strada una nuova forza, quella degli advanced analytics, per determinare predizioni e favorire l'ottimizzazione dei processi. L'impiego di questi strumenti è in progressivo aumento negli ultimi anni e continua ad affermarsi, rivelando una nuova necessità: utilizzare nuove generazioni di algoritmi predittivi e di ottimizzazione robusti, progettati per svolgere analisi di big data. A questa fase della Business Intelligence appartengono le applicazioni analitiche, o business analytics, che raccolgono e consolidano dati provenienti da più fonti e li analizzano fornendo una comprensione più analitica delle attività dell'azienda. Vengono combinate funzioni e componenti separate di BI in un ambiente unico. Il reporting su dati storici può essere integrato con analisi predittive o periodiche. Un ultima fase può essere identificata nella "Smart BI", ovvero nell'agire in anticipo anziché reagire. Si possono prevedere le richieste dell'utente grazie alle sue azioni, anticipando così i suoi bisogni, e ogni cambiamento dell'ambiente in cui opera il sistema si traduce in un intervento per affinare i data model che sono alla base delle analisi.

La Business Intelligence è costituita da due processi principali: l'elaborazione dei dati e l'analisi di essi. Per quanto riguarda l'elaborazione dei dati si ha una modalità tradizionale di uso dei DBMS (database management

system) caratterizzata da un'istantanea del valore corrente dei dati, questi ultimi sono dettagliati e rappresentati in modo relazionale. Le operazioni sono strutturate e ripetitive e l'accesso ai dati avviene in lettura o aggiornando pochi record. Punti critici sono l'isolamento, l'affidabilità e l'integrità. In questa fase le basi di dati hanno una dimensione di circa 100MB-GB. L'analisi dei dati consiste nell'elaborazione dei dati per il supporto alle decisioni ed è caratterizzata da dati di tipo "storico", consolidati ed integrati. Vengono sviluppate applicazioni ad hoc e l'accesso ai dati avviene in lettura a milioni di record. Si hanno interrogazioni di tipo complesso e consistenza dei dati prima e dopo le operazioni di caricamento periodico. In questa fase la dimensione della base di dati è di circa 100GB-TB.

Un processo di supporto alle decisioni aziendali è articolato in diversi stadi. Il primo è sicuramente la raccolta della mole di dati con cui le aziende interagiscono costantemente e la conseguente creazione di una base dati, contenente quindi dati elementari. Il passo seguente è quello di rendere i dati archiviati a disposizione del processo decisionale aziendale: si passa quindi alla costruzione di un "Data Warehouse", una sorta di grande magazzino di dati aziendali riorganizzati in modo funzionale. Dopo aver organizzato i dati in modo strutturato, lo stadio finale è quello di analizzarli tramite strumenti di Business Intelligence e tecniche di Data Mining per accedere ai dati e produrre informazioni utili per l'azienda. In particolare, ad esempio, per analizzare la performance aziendale attuale, prevedere quella futura e utilizzare questi risultati per prendere decisioni chiave per la vita dell'azienda.

## 1.1 Data warehouse

Il data warehouse è una base di dati per il supporto alle decisioni aziendali, mantenuta separatamente dalle basi di dati operative dell'azienda. Al suo interno i dati sono strutturati nel modo seguente:

- orientati ai soggetti di interesse, sono presenti tutti i dati che possono essere usati nel processo di controllo e di decisione e sono raggruppati

per aree, fatti o temi di interesse, destinati poi a chi li utilizza e non a chi li genera;

- integrati e consistenti, i dati aziendali vengono quindi riconciliati in un unico ambiente di analisi eliminando le diverse eterogeneità delle diverse rappresentazioni;
- dipendenti dal tempo (non volatili), il dato è quindi storico e viene caricato periodicamente fuori linea, ovvero una volta memorizzato correttamente l'utente può visionarlo ma non modificarlo.

Il catalogo dati contiene anche una tipologia di dati chiamata metadati, ovvero dei dati sui dati. Ve ne sono di diverso tipo: per la trasformazione e il caricamento (descrivono i dati sorgenti e le trasformazioni necessarie), per la gestione dei dati (descrivono la struttura dei dati presenti nel data warehouse), per la gestione delle query (dati sulla struttura delle query e monitoraggio della loro esecuzione).

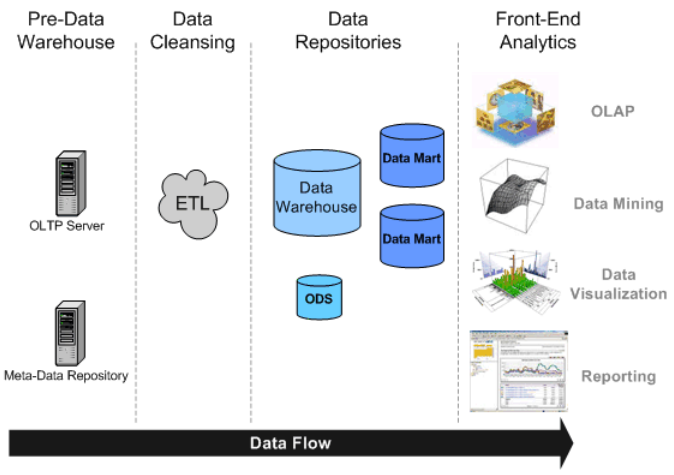


Figura 1.1: Elementi costitutivi di un data warehouse

L'architettura dati può essere costruita in modo da poter accedere a diversi

"magazzini". Uno di questi è il warehouse aziendale che contiene le informazioni sul funzionamento di tutta l'azienda: richiede un processo di modellazione funzionale esteso e la sua progettazione e realizzazione necessitano molto tempo. Per questi ultimi motivi è utile, a volte, utilizzare dati più dettagliati. Si parla in questo caso di data mart: sottoinsieme dipartimentale focalizzato su un settore prefissato. Questo può essere alimentato o dal data warehouse primario o direttamente dalle sorgenti. La sua realizzazione richiede meno tempo ma anche una progettazione attenta, in modo da evitare problemi di integrazione in seguito. Infine, un ultimo "magazzino" di dati è costituito dagli ODS (Operational Data Stores), utilizzati per contenere i dati recenti prima della migrazione al data warehouse.

In base al tipo di "magazzino" impiegato, la progettazione di un data warehouse può seguire due approcci. Il primo è un approccio top-down in cui si ha una realizzazione che fornisce una visione globale e completa dei dati aziendali. Il costo di questa realizzazione è significativo e richiede tempi lunghi. Questo approccio viene utilizzato in genere per analisi e progettazioni complesse. Il secondo è un approccio bottom-up focalizzato separatamente su settori aziendali specifici. È costituito da una realizzazione incrementale del data warehouse, aggiungendo data mart definiti su settori aziendali specifici. In questo caso il costo e il tempo di consegna sono contenuti.

I sistemi di data warehouse possono essere articolati su più livelli e utilizzare più server.

- Server ROLAP (OLAP relazionale): viene utilizzato un DBMS relazionale esteso in cui vi è una rappresentazione compatta di dati sparsi (tutti i dati sono rappresentati come relazioni). Le misure numeriche sono memorizzate nella tabella dei fatti e le dimensioni descrivono il contesto di ogni misura nella tabella dei fatti.

- Server MOLAP (OLAP multidimensionale): i dati sono rappresentati in forma matriciale (multidimensionale) proprietaria e i dati sparsi richiedono compressione. L'informazione viene rappresentata mediante

un insieme di (iper)cubi con tre o più dimensioni le cui celle contengono le misure mentre le dimensioni sono proprio le dimensioni di analisi.

- Server HOLAP (OLAP ibrido).

Le architetture per data warehouse possono essere a due o più livelli, la differenza è che separano in misura diversa i dati in ingresso nel data warehouse dai dati oggetto dell'analisi. Nelle architetture a tre livelli è presente una staging area che si colloca tra le sorgenti di dati e il data warehouse. Questa è un'area di transito che permette di separare l'elaborazione ET dal caricamento nel data warehouse. In questo modo si ha la possibilità di realizzare operazioni complesse di trasformazione e pulizia dei dati. La staging area offre un modello integrato dei dati aziendali ed è il "magazzino" di dati chiamato ODS e già nominato in precedenza. Introduce però ulteriore ridondanza aumentando lo spazio necessario per i dati.

Una volta che il data warehouse è stato progettato e costruito, è necessario alimentarlo e popolarlo attraverso strumenti di back-end. L'alimentazione del data warehouse consiste nel processo di ETL (Extraction Transformation Loading) di preparazione dei dati. Questo è eseguito durante il primo popolamento del data warehouse e durante l'aggiornamento periodico dei dati. Il processo è articolato nei seguenti passaggi:

- estrazione dei dati da sorgenti esterne, consiste nell'acquisizione dei dati dalle sorgenti;
- pulitura, si tratta di operazioni volte al miglioramento della qualità dei dati (errori, dati mancanti o duplicati);
- trasformazioni, ovvero convertire i dati dal formato operazionale a quello del data warehouse (integrazione);
- caricamento e refresh periodico, consiste nella propagazione degli aggiornamenti al data warehouse.

Per quanto riguarda il primo di questi passi, ci sono due modalità di estrazione: statica, che consiste in una fotografia dei dati operazionali ed è eseguita

durante il primo popolamento del DW, e incrementale, in cui avviene la selezione degli aggiornamenti avvenuti dopo l'ultima estrazione ed è utilizzata per l'aggiornamento periodico del DW. La scelta dei dati per l'estrazione è basata sulla loro qualità. Il tipo di estrazione che viene eseguita dipende dai dati operazionali. Se questi sono storicizzati tutte le modifiche vengono memorizzate per un periodo definito di tempo nel sistema OLTP. Se i dati sono semi-storicizzati viene conservato nel sistema OLTP solo un numero limitato di stati. Infine, se i dati sono transitori, il sistema OLTP mantiene solo l'immagine corrente dei dati. Per quanto riguarda la pulitura, i problemi relativi alla qualità dei dati sono dovuti, in genere, a errori di battitura, differenze di formato dei campi ed evoluzione del modo di operare dell'azienda. Ogni problema richiede una tecnica specifica di soluzione. Per errori di battitura o di formato si utilizzano tecniche basate su dizionari, utilizzabili per attributi con dominio ristretto. Tecniche di fusione approssimata sono adatte per il riconoscimento di duplicati o correlazioni tra dati simili. Un'ultima tecnica è costituita dall'identificazione di outliers o deviazioni da business rules. Tuttavia, la strategia migliore è la prevenzione, in modo da rendere più affidabili e rigorose le procedure di data entry OLTP. Il processo di trasformazione richiede una rappresentazione uniforme dei dati operazionali (schema riconciliato). Questo processo può avvenire in due passi: nel primo si ha una passaggio dalle sorgenti operazionali ai dati riconciliati nella staging area (conversioni e normalizzazioni, matching, eventuale filtraggio dei dati significativi), nel secondo si passa dai dati riconciliati al data warehouse (generazioni di chiavi surrogate, generazione di valori aggregati). Infine, riguardo all'ultimo passaggio dell'ETL, per mantenere l'integrità dei dati, si aggiornano in ordine le dimensioni, le tabelle dei fatti e, per ultimo, le viste materializzate e gli indici. Si ha a disposizione una finestra temporale limitata per eseguire gli aggiornamenti e il caricamento richiede proprietà transazionali (affidabilità, atomicità).

Dopo aver popolato il data warehouse si può passare all'analisi dei dati. Esistono diverse operazioni di analisi dei dati:

- calcolo di funzioni aggregate lungo una o più dimensioni tramite il linguaggio SQL;
- operazioni di confronto, queste sono indispensabili per confrontare l'andamento degli affari;
- analisi dei dati mediante tecniche di data mining;
- presentazione, è possibile rappresentare i dati ottenuti da una ricerca mediante diversi tipi di strumenti di rappresentazione, quindi l'operazione è distinta da quella della ricerca;
- ricerca di motivazioni.

È possibile interrogare il data warehouse mediante strumenti di vario tipo: ambiente controllato di query, strumenti specifici di query e generazione rapporti, strumenti di data mining. Approfondiamo ora uno di questi strumenti, ovvero la creazione di un ambiente di query ad hoc. È possibile definire interrogazioni OLAP di tipo arbitrario, progettate al momento dall'utente. Le interrogazioni OLAP consistono nella formulazione di interrogazioni mediante tecniche point and click, le quali generano automaticamente istruzioni SQL. La loro interfaccia è basata sul paradigma dello spreadsheet ed è possibile definire anche interrogazioni complesse. Un documento OLAP consente raffinamenti successivi della stessa interrogazione. Ci sono diverse operazioni di ricerca disponibili in una sessione di lavoro OLAP: roll up, drill down, slice and dice, pivot di tabelle, ordinamento. Queste operazioni possono essere sia combinate tra loro nella stessa query sia eseguite in una sequenza di raffinamenti successivi della stessa query che forma la sessione di lavoro OLAP. L'operazione di roll up consiste nella riduzione di dettaglio dei dati tramite due possibili strade: la riduzione del livello di dettaglio di una delle dimensioni presenti, con l'aumento di livello di una gerarchia, oppure l'eliminazione di una delle dimensioni presenti. L'operazione di drill up è l'esatto opposto di quella di roll up e consiste quindi in un aumento di dettaglio dei

dati. Anche in questo caso si possono seguire due strade: l'aumento del livello di dettaglio di una delle dimensioni presenti, con la riduzione di livello di una gerarchia, oppure l'aggiunta di una nuova dimensione. Lo slice and dice consiste in una riduzione del volume dei dati da analizzare. La selezione di un sottoinsieme da eliminare viene fatta mediante predicati, in particolare lo slice è un predicato di uguaglianza che seleziona una "fetta" e il dice è una combinazione di predicati che seleziona un "cubetto". Il pivot di tabelle è un riorganizzamento dell'orientamento della struttura multidimensionale senza che ci sia una variazione nel livello di dettaglio. Questa operazione permette una visualizzazione più chiara delle stesse informazioni nonostante la rappresentazione dei dati multidimensionali rimanga sotto forma di "griglia".

## 1.2 Data mining

In questo paragrafo verranno presentate le principali tecniche di data mining per l'analisi dei dati. I dati sono un insieme di informazioni contenute in una base di dati o data warehouse. Obiettivo delle tecniche di data mining è l'estrazione di pattern, ovvero di espressioni in un linguaggio opportuno che descrivano in modo breve le informazioni estratte dai dati. I pattern devono avere le seguenti caratteristiche: validità su nuovi dati, novità, utilità, comprensibilità. Il processo di estrazione si compone di diversi passi.

Dopo aver selezionato un sottoinsieme significativo dei dati è necessario preelaborarli, ovvero pulirli, rimuovere il rumore o le eccezioni e gestire i dati mancanti. È possibile ora trasformare i dati, riducendo il numero di variabili da considerare, e passare poi al vero passo di data mining, selezionando il tipo di estrazione e l'algoritmo. In seguito all'interpretazione e alla valutazione dei risultati si arriva alla conoscenza.

Le attività principali della preparazione dei dati sono la pulizia di questi (dati incompleti, rumorosi, riconoscimento di outliers e eccezioni, gestione delle inconsistenze), l'integrazione dei dati, la trasformazione (normalizzazio-

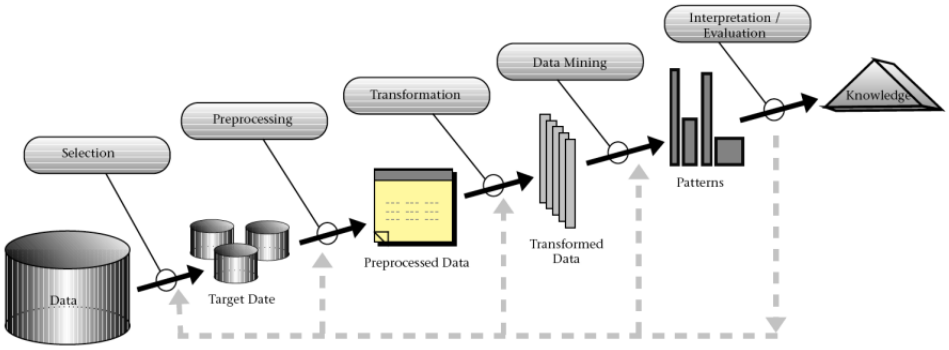


Figura 1.2: Processo di estrazione

ne, aggregazione) e la riduzione dei dati (rappresentazione ridotta in volume). I dati incompleti sono dovuti tipicamente a cause legate al processo di data entry e si possono risolvere usando dei valori speciali, ignorando la tupla con l'informazione mancante, usando come valore il valore medio dell'attributo, etc. I dati rumorosi consistono in errori casuali o in una varianza significativa di una misura e si possono gestire tramite tecniche di clustering o regressione per riconoscere e rimuovere gli outliers e tecniche di discretizzazione. Per ridurre i dati, generando quindi una rappresentazione ridotta in volume che genera risultati analitici simili, si può utilizzare il campionamento (riduzione della cardinalità dell'insieme), la feature selection (riduzione del numero di attributi) o la discretizzazione (riduzione della cardinalità del dominio di un attributo).

Le tecniche di analisi comprendono metodi descrittivi (estrazione di modelli interpretabili che descrivono i dati) e metodi predittivi (utilizzo di alcune variabili conosciute per predire valori sconosciuti o futuri di altre variabili). Le principali tecniche di data mining che verranno qui presentate sono: regole di associazione, classificazione e clustering. L'obiettivo delle regole di associazione è quello di estrarre frequenti correlazioni o pattern da un database transazionale. Il risultato delle regole di associazione è completo (tutte

le regole soddisfano i vincoli alla base dell'estrazione delle regole) e corretto (sono estratte solo le regole che soddisfano i vincoli). Esistono diversi approcci per estrarre le regole: approccio brute-force, principio apriori, algoritmo FP-growth.

Data una collezione di record, chiamata training set, ogni record contiene un insieme di attributi, uno dei quali è la classe. La classificazione consiste nel trovare un modello per l'attributo classe come funzione dei valori degli altri attributi. L'obiettivo è quello di assegnare nuovi record a una particolare classe nel modo più accurato possibile. Un test set viene usato per determinare l'accuratezza del modello. Solitamente il data set che si ha a disposizione viene diviso in training e test set, dove il training set viene usato per la costruzione del modello e il test set per la sua validazione. Vi sono svariate tecniche di classificazione: metodi basati sugli alberi di decisione, metodi basati su regole, reti neurali, support vector machines, etc. Tutte queste tecniche devono però essere accurate (qualità della predizione), efficienti (nel tempo di costruzione del modello e di classificazione), scalabili, robusti (per il rumore e i dati mancanti) e interpretabili.

Un'analisi di cluster consiste nel trovare gruppi di oggetti tali che gli oggetti in un gruppo siano simili (o relazionati) tra di loro e diversi (o non relazionati) dagli oggetti in un altro gruppo. Un clustering è un insieme di cluster e può essere di due tipi: gerarchico, insieme di cluster nidificati organizzati come un albero gerarchico, e partizionale, una divisione dei dati in sottoinsiemi non sovrapposti (clusters) tali che ogni dato è esattamente in un sottoinsieme. Esistono anche diversi tipi di cluster, tra cui i principali sono: cluster ben separati (ogni punto in un dato cluster è più vicino ad ogni punto in quel cluster rispetto agli altri punti non in quel cluster), center-based (ogni oggetto in un cluster è più vicino al "centro" del cluster rispetto al centro degli altri cluster), cluster contigui (un punto in un cluster è più vicino a uno o più punti nel cluster rispetto ad ogni altro punto non nel cluster), density-based (un cluster è un regione densa di punti, separato da altre regioni ad alta densità da regioni a bassa densità). I principali algoritmi di clustering sono

il k-means e le sue varianti, clustering gerarchici (agglomerativi e divisivi) e clustering density-based (DBSCAN).

## 1.3 La Business Intelligence in ambito sanitario

Al giorno d'oggi la nostra epoca si sta evolvendo sempre più verso il digitale. La tecnologia supporta non solo la vita quotidiana degli individui ma anche la gestione e l'organizzazione di ogni tipo di azienda, incluse strutture ospedaliere e studi medici. Negli ultimi anni la Business Intelligence in ambito sanitario è diventata uno strumento fondamentale per la gestione e l'organizzazione di personale medico e pazienti, fondi pubblici e spese che devono assolutamente rientrare nel bilancio annuale. Ma, sostanzialmente, in cosa consiste la Business Intelligence in ambito sanitario? Si tratta di software e strumenti digitali che permettono di gestire, nel miglior modo possibile, un ospedale come se fosse un piccolo studio. Grazie a questi, infatti, è possibile mettere insieme ed analizzare una grande mole di dati economici, clinici, gestionali e di marketing. Le soluzioni di BI consentono di navigare l'insieme di dati presenti all'interno e all'esterno della struttura sanitaria, supportando il personale clinico e amministrativo nella presa di decisioni. Il supporto può essere passivo o attivo. Nel primo caso le intuizioni degli individui vengono verificate grazie a soluzioni di BI, trovando conferma o meno nei dati. Nel secondo caso è la soluzione stessa di BI ad avere "intuizioni", analizzando continuamente e in modo autonomo i dati e scoprendo relazioni statisticamente significative.

L'utilizzo della Business Intelligence in ambito sanitario può portare a grandi benefici. Innanzitutto si può arrivare ad avere una migliore gestione finanziaria. Uno dei problemi più impegnativi per le strutture sanitarie è, infatti, quello della gestione dei costi. Tramite l'utilizzo di software specifici si può tenere sotto controllo il budget, monitorare le spese e gestire il bilancio. Questo viene fatto utilizzando grafici che permettono di capire subito quali sono i problemi della gestione finanziaria in modo da risolverli immediata-

mente. Un altro beneficio consiste in una migliore gestione delle strutture localizzate sul territorio. Tramite la BI è possibile costruire un sistema sanitario più efficiente e sicuro, tenendo sotto controllo sia i servizi erogati dalle singole strutture sia la storia clinica del paziente. Un ultimo vantaggio è l'invio rapido di flussi sanitari alle regioni. La nuova regolamentazione prevede infatti che strutture sanitarie, studi medici e specialisti debbano inviare al Sistema Sanitario una documentazione relativa alle proprie spese. Anche in questo caso, tramite software specifici, è possibile trasmettere una mole di informazioni molto ampia, proveniente anche da piattaforme diverse tra loro ma facilmente trasmissibili.

# Capitolo 2

# Cenni di statistica

In questo capitolo vengono esposti i concetti di statistica alla base del lavoro che sarà presentato nell'ultimo capitolo. Prima di concentrarsi sulle due principali nozioni statistiche utilizzate, ovvero i test di ipotesi e la regressione logistica, vengono di seguito illustrati e spiegati nel dettaglio i grafici utilizzati.

I principali grafici impiegati per rappresentare i dati sono l'istogramma e il boxplot. Un esempio di istogramma è il seguente:

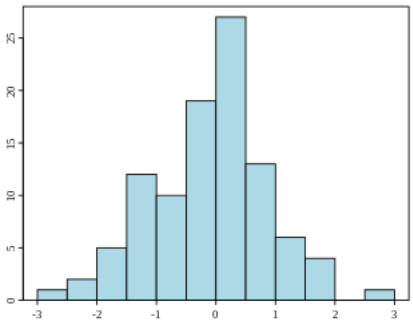


Figura 2.1: Esempio di istogramma

Un istogramma è un diagramma che fornisce una rappresentazione di un insieme di dati statistici mediante un grafico a barre. Gli istogrammi possono essere rappresentati mediante barre orizzontali o verticali e consentono di

rappresentare i dati attraverso rettangoli di uguale base ed altezza differente a seconda dei dati considerati. Più precisamente, dato un insieme di nodi arbitrario $t_0 < \min_i x_i < t_1 < \ldots < \max_i x_i < t_n$, l'istogramma è una funzione a gradini tale che il suo valore in $x$ quando $t_{i-1} < x \leq t_i$ è pari a

$$h(x) = \frac{\sum_{k=1}^{N}(x_k \in (t_{i-1}, t_i])}{N \times (t_i - t_{i-1})}.$$

L'istogramma $h(\cdot)$ è quindi una funzione costante su ogni intervallo $(t_{i-1}, t_i]$, tale che il suo integrale tra $t_{i-1}$ e $t_i$, cioè l'area del rettangolo tra $t_{i-1}$ e $t_i$, è uguale alla frequenza relativa delle osservazioni che cadono in quell'intervallo e l'integrale totale dell'istogramma è uguale a 1. La scelta del numero di classi in cui suddividere l'intervallo in cui variano i dati non è indifferente: troppo poche appiattiscono il grafico fino a renderlo insignificante; troppe classi introducono tra le barre oscillazioni eccessive, che potrebbero distruggere l'eventuale "regolarità" dell'istogramma. Questa scelta varia in base al tipo di dati che si ha a disposizione.

In statistica descrittiva, il boxplot è un metodo per rappresentare una distribuzione statistica nel modo seguente:

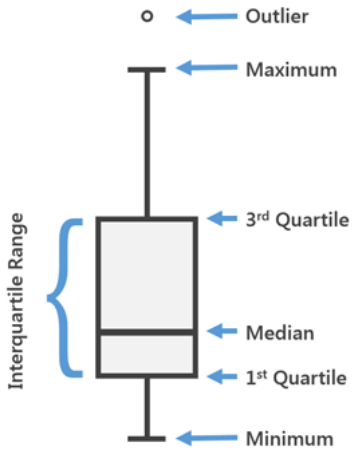


Figura 2.2: Esempio di boxplot

La linea interna alla scatola rappresenta la mediana della distribuzione, men-

tre le linee estreme del box rappresentano il primo ed il terzo quartile (rispettivamente, quartile inferiore o $Q_1$ e quartile superiore o $Q_3$). La distanza tra il terzo ed il primo quartile, distanza interquartilica, è una misura della dispersione della distribuzione. Il 50% delle osservazioni si trovano comprese tra questi due valori. Se l'intervallo interquartilico è piccolo, tale metà delle osservazioni si trova fortemente concentrata intorno alla mediana; all'aumentare di questa distanza aumenta la dispersione del 50% delle osservazioni centrali intorno alla mediana. Le distanze tra ciascun quartile e la mediana, inoltre, forniscono delle informazioni riguardo alla forma della distribuzione: se una distanza è diversa dall'altra allora la distribuzione è asimmetrica. Le linee che si allungano dai bordi del box, chiamati baffi, individuano gli intervalli in cui sono posizionati i valori rispettivamente minori di $Q_1$ e maggiori di $Q_3$; i punti estremi dei baffi evidenziano i valori adiacenti. Se si indica con $r = Q_3 - Q_1$ la differenza interquartilica, il valore adiacente inferiore è il valore più piccolo tra le osservazioni che risulta maggiore o uguale a $Q_1 - 1.5r$. Il valore adiacente superiore, invece, è il valore più grande tra le osservazioni che risulta minore o uguale a $Q_3 + 1.5r$. Pertanto, se gli estremi della distribuzione sono contenuti tra $Q_1 - 1.5r$ e $Q_3 + 1.5r$, essi coincideranno con gli estremi dei baffi, altrimenti come estremi verranno utilizzati i valori $Q_1 - 1.5r$ e $Q_3 + 1.5r$. I valori esterni a questi limiti (esterni rispetto ai valori adiacenti) vengono chiamati valori anomali o outliers e segnalati individualmente nel boxplot per evidenziare meglio la loro presenza e posizione. Questi valori rappresentano infatti una "anomalia" rispetto alla maggior parte dei valori osservati e, di conseguenza, è importante identificarli per poterne analizzare le caratteristiche e le eventuali cause che li hanno determinati. Questi forniscono ulteriori informazioni sulla dispersione e sulla forma della distribuzione. Quando i valori adiacenti, superiore e inferiore, coincidono con gli estremi della distribuzione non comparirà alcun valore fuori limite. I valori adiacenti inferiore e superiore forniscono informazioni sulla dispersione e sulla forma della distribuzione ed anche sulle code della distribuzione. Nel caso di una distribuzione normale, ad esempio, nel boxplot le distanze tra ciascun

quartile e la mediana saranno uguali, così pure avranno uguale lunghezza le linee che si allungano dai bordi della scatola, ovvero i baffi. Per concludere, i boxplot sono non parametrici: mostrano la variazione nei campioni di una popolazione statistica senza fare nessuna assunzione della distribuzione statistica sottostante.

## 2.1 Test di ipotesi: Fisher e Chi quadro

Il test esatto di Fisher è un test di significatività statistico usato nelle analisi delle tabelle di contingenza. Questo test non parametrico è fondamentale per l'interpretazione statistica di molti dati biologici. Fa parte di una classe di test esatti, così chiamati in quanto la significatività della deviazione da un'ipotesi nulla, il p-value, può essere calcolata esattamente anziché fare affidamento su un'approssimazione che diventa esatta nei limiti (dato che la dimensione del campione cresce all'infinito) come in molti test statistici. Il test è utile per dati categorici che derivano dalla classificazione degli oggetti in due modi differenti. È usato per esaminare la significatività dell'associazione (contingenza) tra i due tipi di classificazione. In seguito alla suddivisione dei dati in due popolazioni differenti secondo un dato criterio, si può formulare l'ipotesi nulla del test nel seguente modo:

$H_0$: non esiste alcuna differenza tra le popolazioni suddivise in base al parametro considerato e gli eventi osservati sono dovuti al caso senza che ci sia correlazione tra questi (le popolazioni sono omogenee)

La maggior parte degli utilizzi del test di Fisher coinvolgono una tabella di contingenza $2 \times 2$ dove sono riportate le frequenza assolute (a, b, c, d) di quattro sottogruppi suddivisi in base a due criteri (nel caso di studio che verrà presentato nell'ultimo capitolo questi criteri saranno esente o non esente e patologico o normale). La tabella può essere anche trasposta senza che il risultato finale non cambi. Di seguito viene riportata un esempio di tabella di contingenza.

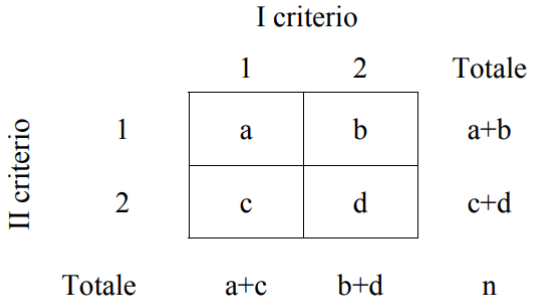


Figura 2.3: Esempio tabella di contingenza

Fisher dimostrò che la probabilità di ottenere i valori nelle celle segue la variabile casuale ipergeometrica ed è data da:

$$p = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}} = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!}$$

dove $\binom{n}{k}$ è il coefficiente binomiale e il simbolo ! indica l'operatore fattoriale. La formula sopra dà le probabilità ipergeometriche esatte di osservare i valori $a, b, c, d$ nel caso fosse vera l'ipotesi nulla formulata sopra.

Generalmente, per decidere se accettare o meno l'ipotesi nulla di un test di ipotesi, si utilizza il p-value. Per calcolarlo occorre ricavare una seconda tabella da quella originaria azzerando il valore della cella che contiene il valore più basso. I valori delle altre celle dovranno essere cambiati di conseguenza in modo da mantenere costanti i totali parziali. A questo punto, utilizzando la formula esposta sopra e dimostrata da Fisher, si otterranno due probabilità $p_0$ e $p_1$, rispettivamente per la prima e la seconda tabella. Per ottenere la significatività dei dati osservati è necessario sommare questi due valori insieme ottenendo $p - value = p_0 + p_1$. Poiché, tuttavia, questo calcolo è molto laborioso, si utilizzano generalmente dei software applicativi per la statistica, come R, per il calcolo. Il p-value può essere interpretato come la somma delle evidenze provate dai dati osservati per l'ipotesi nulla. Più è piccolo il valore di quest'ultimo, più è grande la prova per rifiutare l'ipotesi nulla.

Per grandi campioni (nessuna cella deve avere un valore inferiore a 5) è meglio utilizzare il test chi-quadro anziché quello di Fisher. Questo test è esatto solo asintoticamente per dimensioni molto grandi dei campioni e il valore di significatività che fornisce è solo un'approssimazione, poiché la distribuzione campionaria della statistica del test che viene calcolata è solo approssimativamente vicina alla distribuzione teorica chi-quadro. Per questi motivi questa approssimazione è inadeguata quando i campioni sono piccoli e in questi casi si utilizza il test esatto di Fisher che è sempre esatto. Nel test chi-quadro si utilizza sempre una tabella di contingenza come quella di Figura 4.7 e l'ipotesi nulla è la stessa di quella formulata per il test di Fisher. Per eseguire il test occorre calcolare la satistica test $\chi^2$ per la differenza tra due proporzioni. Per fare questo occorre quantificare per ogni cella la differenza tra i dati osservati e quelli attesi e sommare poi questi quattro valori. Si ha quindi la seguente formula:

$$\chi^2 = \sum_{tutte\ le\ celle} \frac{(frequenza \quad osservata - frequenza \quad attesa)^2}{frequenza \quad attesa}.$$

La variabile test chi-quadro è distribuita secondo una variabile casuale Chi-quadro con $(g-1)$ gradi di libertà ($\chi^2_{g-1}$), dove $g = (numero\,di\,righe - 1)(numero\,di\,colonne - 1)$ se si guarda alla tabella di contingenza (nella Figura 4.7 $g = 1$). Fissato $\alpha$, l'ipotesi nulla dovrà essere rifiutata se il valore osservato della statistica $\chi^2$ è maggiore del valore critico $\chi^2_U$ di una distribuzione $\chi^2$ con $(g-1)$ gradi di libertà.

Anche in questo caso, come nel test di Fisher, si può guardare anche al p-value (misura di quanto i dati sono coerenti con l'ipotesi nulla) per decidere se rifiutare o meno il test di ipotesi. In genere, se questo è inferiore a 0.05, l'ipotesi nulla viene rifiutata.

## 2.2 Regressione logistica

La regressione logistica è uno degli approcci per predire le variabili risposte qualitative, anche dette categoriche; questi processi sono conosciuti come

classificatori. Predire una risposta qualitativa per un'osservazione può essere riferito a classificare quell'osservazione, poiché riguarda l'assegnare l'osservazione a una categoria o una classe. In generale, non c'è un modo naturale per convertire una risposta qualitativa con più di due livelli in una risposta quantitativa che è pronta per la regressione lineare. Per questo si utilizzano i metodi di classificazione.

Consideriamo il seguente modello:

$$Y = \beta_0 + \beta_1 X$$

dove $Y$ è la variabile risposta qualitativa, $X$ è il predittore e $\beta_0$ e $\beta_1$ sono i coefficienti. La regressione logistica, anziché modellare direttamente la risposta $Y$, modella la probabilità che $Y$ appartenga a una particolare categoria. Occorre trovare un modo per modellare la relazione tra la probabilità $p(x) = \Pr(Y = 1|X)$ e $X$. Per modellare la probabilità bisogna usare una funzione che produce degli output tra 0 e 1 per tutti i valori di X. Per questo modello di classificazione si usa la funzione logistica:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

La funzione logistica produrrà sempre una curva a forma di S e quindi, indipendentemente dal valore di $X$, si otterrà sempre una predizione sensibile. Per fittare il modello si utilizza un metodo chiamato massima verosimiglianza.

Dopo aver manipolato un po' la funzione logistica, si trova che

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X}.$$

La quantità sopra è chiamata odds e può assumere qualsiasi valore tra 0 e $\infty$. Valori dell'odds vicini a 0 e $\infty$ corrispondono, rispettivamente, a probabilità molto basse e molto alte. L'inversa sarà data da

$$p(X) = \frac{\text{odds}(X)}{1 + \text{odds}(X)}.$$

Un'altra trasformazione molto usata è il log-odds, o logit, che si ottiene applicando il logaritmo a entrambe le parti della formula dell'odds:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

Questa trasformazione è la più usata per "spiegare" una probabilità in termini di predittori. Il modello di regressione logistica ha un logit che è lineare in $X$. Nel modello di regressione logistica aumentare $X$ di una unità cambia il log-odds di $\beta_1$, o equivalentemente moltiplica l'odds di $e^{\beta_1}$. Tuttavia, poiché la relazione tra $p(X)$ e $X$ non è una linea retta, $\beta_1$ non corrisponde al cambiamento in $p(X)$ associato con un aumento di un'unità in $X$. La quantità di cambiamento in $p(X)$, dovuta alla variazione in $X$ di una unità, dipenderà dal valore corrente di $X$. Invece, indipendentemente dal valore di $X$, se $\beta_1$ è positivo allora l'aumento di $X$ sarà associato ad un aumento di $p(X)$, se $\beta_1$ è negativo allora l'aumento di $X$ sarà associato ad una diminuzione di $p(X)$.

I coefficienti $\beta_0$ e $\beta_1$ sono sconosciuti e devono essere stimati sulla base dei dati disponibili. Per fare questo viene utilizzato il metodo della massima verosimiglianza. L'intuizione di base dietro all'utilizzo di questo metodo per fittare un modello di regressione logistica è la seguente: si ricerca una stima per $\beta_0$ e $\beta_1$ in modo che la probabilità predetta $\hat{p}(x_i)$ per ogni individuo corrisponde il più strettamente possibile allo stato osservato dell'individuo. Questa intuizione può essere formalizzata usando una formula matematica, la funzione di verosimiglianza:

$$\mathcal{L}(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

Le stime $\hat{\beta}_0$ e $\hat{\beta}_1$ sono scelte per massimizzare questa funzione di verosimiglianza. È possibile, inoltre, stimare l'accuratezza dei coefficienti stimati calcolando i loro errori standard.

Una volta stimati i coefficienti, risulta facile fare delle previsioni per la probabilità. Supponendo infatti di avere un particolare valore per $X$ ($X = x_1$) basta applicare la seguente formula:

$$\hat{p}(x_1) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1}}.$$

Considerando ora il problema di predire una risposta binaria usando predittori multipli, questo si affronta in maniera analoga al problema con un solo predittore discusso finora. Le formule per il calcolo della funzione logistica e del log-odds si possono infatti generalizzare nel seguente modo:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p.$$

Come discusso prima, anche in questo caso si utilizza la massima verosimiglianza per stimare $\beta_0, \beta_1, \ldots, \beta_p$.

Quando si affrontano problemi di questa tipologia i dati si possono presentare in due modi differenti. Questi possono essere aggregati o non aggregati: nel primo caso gruppi di osservazioni sono sostituiti da statistiche sommarie basate su queste osservazioni. Prendendo come esempio i dati riguardanti uno studio clinico, nel caso in cui questi siano aggregati si avrebbe una riga per trattamento e, per ogni riga, il numero di pazienti e il numero di successi. Nel caso di dati non aggregati si avrebbe invece una riga per ogni paziente e, per ogni paziente, che tipo di trattamento è stato utilizzato e se questo ha avuto successo o meno.

Se si considera il caso di dati aggregati, la variabile risposta diventa: (numero di successi, numero di fallimenti). In questo caso è possibile fare le seguenti osservazioni per quanto riguarda l'odds:

- se la probabilità di successo è uguale a quella di fallimento, allora odds(successo)=1;
- se la probabilità di successo è minore di quella di fallimento, allora odds(successo)<1;
- se la probabilità di successo è maggiore di quella di fallimento, allora odds(successo)>1.

Oltre alle formule enunciate precedentemente, per questi modelli si calcola anche l'odds-ratio (OR) di una categoria B (ad esempio un tipo di trattamento) contro una categoria A:

$$OR_{BA} = \frac{\frac{p_B}{(1-p_B)}}{\frac{p_A}{(1-p_A)}}$$

dove $p_A$ e $p_B$ sono le probabilità dell'evento $A$ e $B$ rispettivamente.

Per verificare la qualità dei modelli, sia nel caso di dati aggregati che non aggregati, si utilizza la formula della devianza per il confronto di due modelli

$$\mathrm{D}(y) = 2\log\left[\frac{\mathcal{L}(modello_1, y)}{\mathcal{L}(modello_2, y)}\right] =$$
$$= 2[\log\mathcal{L}(modello_1, y) - \log\mathcal{L}(modello_2, y)] \sim \chi^2(\mathrm{df}_2 - \mathrm{df}_1)$$

dove con $\mathrm{df}_i$ si intende i gradi di libertà del modello i-esimo e $\mathcal{L}$ è la funzione di verosimiglianza. Se la devianza, che si distribuisce secondo una variabile chi-quadro con $(\mathrm{df}_2 - \mathrm{df}_1)$ gradi di libertà, è grande significa che è statisticamente significativa e che il modello 1 è migliore.

In particolare, per i modelli considerati finora, si calcola la devianza nulla e residua:

$$\text{Devianza Nulla} = \mathrm{D}_0(y) =$$
$$= 2[\log\mathcal{L}(Modello\ Saturato, y) - \log\mathcal{L}(Modello\ Nullo, y)]$$
$$\sim \chi^2(\mathrm{df\_Sat} - \mathrm{df\_Null})$$

$$\text{Devianza Residua} = \mathrm{D}_R(y) =$$
$$= 2[\log\mathcal{L}(Modello\ Saturato, y) - \log\mathcal{L}(Modello\ Proposto, y)]$$
$$\sim \chi^2(\mathrm{df\_Sat} - \mathrm{df\_Res})$$

Df_Sat, df_Null e df_Res sono, rispettivamente, i gradi di libertà del modello saturato, nullo e proposto. Il modello saturato è un modello che fitta perfettamente i dati e ha un parametro per ogni osservazione, si hanno quindi un numero di predittori uguale al numero di dati a disposizione (Df_Sat =

$n$, con $n$ = numero di osservazioni). Il modello nullo, invece, assume l'esatto opposto, ovvero che si ha un parametro (quindi un solo predittore: l'intercetta) per tutti i dati (df_Null = 1). Il modello proposto presuppone che i dati si possono "spiegare" con $p$ parametri, con $1 < p < n$, e si hanno quindi $p$ predittori (df_Res = $p$). Se la devianza nulla è molto piccola significa che il modello nullo è migliore del modello saturato e che spiega bene i dati. Se la devianza residua è molto piccola significa che il modello proposto è migliore di quello saturato e che fitta bene i dati.

# Capitolo 3

# L'azienda e Knowage

## 3.1 Engineering Ingegneria Informatica SpA


Figura 3.1: Logo Engineering Ingegneria Informatica SpA

Engineering Ingegneria Informatica SpA è un'azienda nata a Padova nel 1980 (quando l'informatica in Italia muoveva ancora i primi passi) da Cerved, la società di informatica del sistema camerale italiano, oggi Infocamere e grazie a un'operazione di management buy out. Accanto ai manager azionisti, si avvicendano nel tempo soci industriali e finanziari con partecipazioni di minoranza e nel 2000 si ha l'ingresso in Borsa della capogruppo nel segmento FTSE Italia STAR per i titoli con i più alti requisiti patrimoniali. Grazie all'arrivo di nuovi capitali provenienti dal mercato è stata finanziata la crescita delle attività, per linea interna e per acquisizioni, in un mercato sempre più globale e competitivo.

Engineering ha conquistato la leadership nei servizi ICT sul mercato italiano e internazionale con 50 sedi in totale: è presente sul territorio nazionale in tutte le regioni e nei principali comuni, con filiali estere in Europa (Germania, Spagna, Belgio e Repubblica di Serbia), Sud America (Brasile e Argentina) e Stati Uniti. Ha oltre 9000 specialisti che ogni giorno sviluppano progetti, erogano servizi, integrano suite di prodotti e soluzioni in architetture e sistemi operativi e possiede un portafoglio ricavi consolidato nel 2016 di circa 934,6 milioni di Euro. L'azienda ha più di 1000 clienti italiani e esteri con una presenza consolidata su tutti i mercati verticali e opera attraverso 4 business unit: Pubblica Amministrazione e Sanità, Telco & Utilities, Industria e Servizi, Finanza. Queste ultime sono supportate da centri di competenza trasversali rispetto alle business unit e dalla Direzione Ricerca & Innovazione che, con circa 250 risorse, ha il doppio ruolo di promuovere la ricerca sul software a livello internazionale e trasferire l'innovazione al ciclo produttivo delle strutture di business. Ricoprendo un ruolo di primaria importanza nel mercato dell'outsourcing e del cloud computing, Engineering opera attraverso un network integrato di 4 data center: Pont-Saint-Martin, Torino, Milano e Vicenza. Ha un sistema di servizi e un'infrastruttura che garantiscono i migliori standard tecnologici, qualitativi e di sicurezza. Un grande punto di forza ed esclusività nel panorama nazionale è la Scuola di IT & Management "Enrico Della Valle": con 200 docenti certificati e 363 corsi disponibili offre 20000 giornate di formazione tecnica, metodologica e di processo all'anno.

L'obiettivo primario di Engineering è quello di progettare e realizzare architetture innovative che orientano e supportano i modelli di business e di servizio di tutte le organizzazioni e su tutti i mercati. Questo è realizzato tramite la promozione dell'innovazione nelle organizzazioni, nei processi e nei servizi.

## 3.2 Knowage

Knowage è una nuova suite open source di business analytics per le aziende del futuro. Sviluppata all'interno di Engineering da un gruppo di esperti, viene utilizzata per la gestione di diversi progetti. È la suite ideale per valorizzare il proprio patrimonio di dati, anche in relazione a sorgenti informative esterne ed eterogenee, e coglie pienamente il significato dei big data.


Figura 3.2: Slogan Knowage

Knowage permette di costruire analisi avanzate e visualizzazioni personalizzate sui propri dati in completa autonomia, permettendo di unire, tramite tecniche di mash-up e data federation, le sorgenti tradizionali e strutturate con quelle più innovative dell'ecosistema Hadoop e del mondo NoSQL. Grazie alla possibilità di legare fonti informative differenti è possibile applicare metodologie all'avanguardia per analisi non solo descrittive ma anche diagnostiche, predittive e prescrittive; tutto ciò può essere fatto attraverso strumenti di data/text mining basati su algoritmi di machine learning e statistiche avanzate. Knowage è una suite fruibile da diverse tipologie di utenti in quanto offre un'ampia gamma di strumenti modulabili sulle proprie esigenze e sulle competenze: dai decision-maker che hanno bisogno di informazioni chiare, precise e puntuali, agli utenti più operativi inclini all'analisi in autonomia, ai data scientist che operano per cicli incrementali di ipotesi e verifica.

È costituita da diversi moduli specializzati per ambito analitico e combinabili tra loro per garantire una completa e flessibile copertura dei requisiti utente; questo anche nella modalità di fruizione, essendo disponibile in moda-

lità on-premise, as-a-service ed in cloud. Gli 8 moduli di cui è dotata la suite vengono presentati di seguito, ognuno dotato di un ampio set di capacità analitiche.

- **Big Data (BD)**: non è solo una questione di volume ma anche di operare con diversi tipi di dati e sorgenti, combinando dati aziendali strutturati con dati esterni multi strutturati. Questo modulo permette di lavorare con sorgenti big data e tradizionali, federando i data set in modo da costruire diverse analisi quali report statici, mappe, cockpit interattivi, etc. La federazione dei dati permette di lavorare con diverse sorgenti di dati allo stesso tempo, combinando data set eterogenei in un modello comune. È possibile dichiarare le corrispondenze logiche tra i dati o chiedere a Knowage di proporre quelle auto individuate. Il modello federato può poi essere usato in ogni documento analitico come qualsiasi altro data set. All'interno di questo modulo l'utente può inoltre esplorare liberamente i propri dati usando un generatore di query drag & drop o dando uno sguardo immediato grazie a visualizzazioni avanzate. Con i big data l'approccio della visualizzazione avanzata diventa centrale in quanto l'utente finale ha bisogno di essere guidato attraverso eventi rilevanti senza però essere sommerso dai dettagli di questi eventi.

- **Performance Management (PM)**: riguarda l'abilità di misurare e valutare le perforamnce di business. Questo non richiede solo le misurazioni degli obiettivi e dei targets in base a specifiche soglie e da diverse prospettive, ma anche la capacità di reagire prontamente a eventi critici e problemi inaspettati, conseguenze di specifici allarmi e notifiche. Knowage PM, infatti, fornisce segnali di allarme per avvertire il prima possibile gli utenti tecnici e di business che un KPI rilevante ha sforato le sue soglie. Il modulo dà l'opportunità di lavorare con sorgenti di dati tradizionali o dati esterni, in modo da stabilire KPIs, obiettivi e targets fino a costruire documenti composti degli obiettivi. I KPIs sono indi-

catori chiave di performance che permettono all'utente di analizzare le performance di business.

- **Predictive Analysis (PA)**: riguarda l'abilità di realizzare processi avanzati usando tecniche di data mining per scopi previsionali e prescrittivi. I data scientist possono scrivere complessi modelli analitici che un normale utente può poi utilizzare nelle sue valutazioni giornaliere e nel suo lavoro, grazie a un uso semplice e parametrico che fa sì che tutti possano produrre analisi avanzate. Occorre essere in grado di simulare azioni e valutare i loro effetti su diverse attività. Questo modulo permette di lavorare con sorgenti di dati tradizionali o file esterni, processandoli con algoritmi avanzati scritti usando R/Spark e altri linguaggi di scrittura. Per scopi what-if, il modulo fornisce una soluzione basata sugli OLAP che permette la simulazione interattiva del processo su misure e dimensioni differenti. I documenti OLAP consentono di esplorare e navigare i dati su diversi livelli di dettaglio e da diverse prospettive tramite opzioni di drill-down, drill-across, slice-and-dice e drill-through. È possibile simulare nuovi valori su ogni livello e valutare l'effetto per ogni dimensione, comparando diversi scenari e versioni.

- **Open Data (OD)**

- **Enterprise Reporting (ER)**: si riferisce al rilascio di informazioni puntuali da un insieme diversificato di utenti, al tempo giusto. Questo approccio si focalizza su una sicurezza basata sul profilo degli utenti che permette loro di accedere a informazioni significative come dati certificati, nel formato appropriato. Questo modulo permette di lavorare con sorgenti di dati tradizionali per produrre report statici utilizzabili con un ampio insieme di parametri e che producono diversi output. I report presentano le informazioni tramite uno schema strutturato e facile da leggere; quest'ultimo può essere tradizionale (tramite l'utilizzo di tabelle e chart) o più creativo nella forma delle infografiche. I report possono essere prodotti on-line, leggendo i dati quando l'utente

li carica, o off-line, per averli pronti da leggere con un click. Questi possono poi essere esportati e distribuiti come PDF, CSV, XLS, RTF o XLSX ed è anche possibile definire il sistema per produrli e distribuirli in maniera massiva, organizzando il giusto output per il giusto utente.

- **Smart Intelligence (SI)**: permette un facile accesso a dati strutturati usando delle analisi pre costruite, garantendo così totale controllo sui propri dati grazie ad avanzate tecniche ad-hoc e auto gestite. Questo modulo offre l'opportunità di lavorare con sorgenti di dati tradizionali, anche combinandoli tra loro, per costruire analisi quali cruscotti, reports e analisi multidimensionali (OLAP). I cockpit interattivi consentono l'esplorazione dei dati e la creazione di self-service report. Si possono combinare dati e widget in modo che anche un utente aziendale possa costruire la propria dashboard in pochi minuti. Supporta lo staff IT a gestire l'ambiente aziendale con metadati multi-tenant e complessi. Un metamodello è una vista aziendale logica sui sistemi tecnici di memoria che organizzano i dati. Una volta costruito da un utente tecnico che ben conosce i dati e le regole di visibilità, il metamodelllo può essere liberamente usato dagli utenti finali per interrogare i dati senza che questi conoscano la loro struttura. Il modulo permette inoltre all'utente finale di interrogare liberamente i propri dati e produrre analisi e visualizzazioni.

- **Location Intelligence (LI)**: si riferisce al poter plottare dati aziendali su una mappa, uno spazio, uno schema o una figura vettoriale. Questo permette di dare uno sguardo immediato ai dati, grazie anche a tecniche di mash-up, senza il bisogno di spostarli tra sistemi GIS e strumenti di data warehouse. Questo modulo consente di lavorare con sorgenti di dati tradizionali e dati spaziali con una relazione reale tra questi, in modo da produrre mappe dinamiche tramite WMS/WFS standard o mappe statiche e figure in formato SVG.

- **Embedded Intelligence (EI)**: fa sì che Knowage sia aperto. Può

essere collegato a soluzioni di terze parti sotto i termini e le condizioni della licenza AGPL v3. Il modulo permette di lavorare con servizi Java API, Javascript API, REST e altre interfacce tecniche per offrire supporto SSO o l'integrazione con depositi utenti esterni.

All'interno della suite sono definiti tre diversi ruoli: amministratore, utente finale e sviluppatore. I ruoli rappresentano delle categorizzazioni di gruppi di utenti e accordano a ogni utente diversi diritti e criteri di visibilità sui documenti e sui dati, in base al loro profilo di business. L'amministratore è colui che dispone di più funzionalità all'interno di Knowage e, oltre a poter creare nuovi utenti, può settare i diritti di ognuno e decidere cosa i diversi utenti sono in grado di vedere.

Knowage è disponibile in due diverse versioni: Community Edition (CE) ed Enterprise Edition (EE). L'edizione CE è open source ed è questo uno dei più grandi punti di forza di Knowage. Open source è visto come sinonimo di innovazione ed è proprio per questo che la versione CE non è una versione "giocattolo" o di prova con restrizioni rilevanti. Include l'intero insieme di funzionalità analitiche e garantisce una piena esperienza per l'utente finale. Questa versione è liberamente scaricabile sul sito ed è anche possibile consultare un manuale utente in cui vengono spiegate tutte le principali funzionalità. L'edizione EE è invece fornita direttamente da Engineering Group grazie a una sottoscrizione annuale. Le aziende hanno bisogno di applicazioni a scala industriale che si adattino a standard di qualità e sicurezza, idonee per diversi tipi di utente, garantite per utilizzi strategici e scenari critici. I software industriali processano aspetti molto importanti per ogni strumento all'interno dell'azienda. Per questi motivi Knowage EE rende più facile la realizzazione di alcune funzioni amministrative quali l'installazione, la migrazione, l'aggiornamento, la manutenzione e il monitoraggio. Questa versione offre alcune caratteristiche avanzate e servizi che vengono assicurati essere efficaci e assicura il proprio ritorno degli investimenti. Alcuni dei servizi offerti da questa edizione sono i seguenti: documentazione professionale, servizi di manutenzione, accesso all'area e agli strumenti del cliente, accesso prioritario

alle modifiche e al fissaggio di bachi, accesso prioritario a nuovi rilasci. In aggiunta a questa offerta commerciale è possibile includere altri servizi quali consulenze tecniche, formazione ad-hoc, supporto alla migrazione, etc.

Sul sito ufficiale di Knowage sono disponibili diverse risorse anche per gli utenti che hanno scelto di scaricare la versione CE. È possibile consultare alcuni video di demo per, ad esempio, avere una panoramica dei principali documenti analitici forniti dalla suite (mappe, report, OLAP, cockpit interattivi, etc.), per avere una guida per la creazione di cockpit, metamodelli, federazione dei dati, etc. Una comunità attiva è la forza di un prodotto open source. Per questo sono disponibili strumenti di supporto liberamente utilizzabili senza però la garanzia di un supporto tecnico. Il primo di questi strumenti è la pagina Q&A che permette ai vari utenti di entrare in contatto tra di loro e con lo staff tecnico di Knowage per risolvere eventuali dubbi o difficoltà. È inoltre possibile, grazie al secondo strumento di supporto, riportare dei bachi, contribuendo così a fissare quelli già riportati, e suggerire nuovi requisiti e nuove funzionalità. Infine, è possibile scaricare liberamente la documentazione di Knowage sull'apposito sito, che include il manuale utente con la descrizione delle principali funzionalità di Knowage. Vi è anche la possibilità di partecipare a seminari web gratuiti sia per avere una panoramica comprensiva di Knowage sia, per gli utenti più esperti, per rimanere aggiornati sulle principali novità.

### 3.2.1 Altri software in ambito BI

Esistono svariati software di business intelligence che aiutano le aziende ad organizzare e analizzare i dati per prendere migliori decisioni. Il mercato della BI sta crescendo velocemente a causa dell'aumento costante di dati da analizzare. I software BI possono essere divisi in tre grandi categorie di applicazione: strumenti per la gestione dei dati, applicazioni per la scoperta dei dati e strumenti di reporting (incluse dashboards e software di visualizzazione). Di quale strumento BI si abbia bisogno dipende da come i dati sono correntemente gestiti e come li si vuole analizzare.

Nel seguito verranno presentati alcuni BI software.

- **Pentaho**: aiuta le aziende a prendere decisioni sui dati tramite una piattaforma per l'analisi e l'integrazione di questi. La piattaforma include ETL, analisi di big data, visualizzazioni, dashboards, reporting, data mining e analisi predittive. È disponibile sia una versione Community edition (libera e open source) che una versione Enterprise edition (per la quale si deve comprare una sottoscrizione). Per avere realmente dei vantaggi dall'utilizzo di questo software è però necessario avere a disposizione qualcuno che conosce come programmare.

- **Qlik View**: prodotto di scoperta dei dati per la creazione di applicazioni di analisi guidate e cruscotti fatti su misura per le attività commerciali. Il software permette agli utenti di svelare dati approfonditi e relazioni tra varie sorgenti. Offre inoltre esplorazioni guidate, scoperte e analisi collaborative per condividere le intuizioni. Inoltre, il programma consente agli utenti di costruire e sviluppare applicazioni analitiche senza aver bisogno di saper sviluppare in modo professionale.

- **Qlik Sense**: piattaforma di business intelligence e analisi visiva che supporta una serie di casi d'uso, inclusi app di analisi e cruscotti guidati, analisi personalizzate e incorporate e visualizzazioni self-service; il tutto all'interno di una struttura scalabile e governata. Sono disponibili tre diverse edizioni: Qlik Sense Desktop, Enterprise e Cloud. Il sistema offre visualizzazioni e scoperta di dati per individui e team.

- **Jaspersoft**: suite self-service scalabile idonea dalle aziende piccole a quelle di livello enterprise. Le caratteristiche di drag and drop consentono agli utenti di creare cruscotti e report. Questo sistema include anche un report scheduler che fa sì che gli utenti possano gestire la distribuzione dei report attraverso l'azienda. Le opzioni di visualizzazioni includono mappe interattive e cruscotti che permettono agli utenti di inseguire le metriche aziendali.

- **Tableau**: soluzione di analisi integrata che aiuta ad analizzare dati chiave di business e a generare intuizioni significative. Aiuta le aziende a collezionare dati da sorgenti multiple. Le analisi visive realistiche e i cruscotti interattivi ammettono datasets slicing & dicing per generare intuizioni rilevanti ed esplorare nuove opportunità. Gli utenti possono creare mappe interattive e analizzare dati attraverso regioni, territori, demografia e molto altro. Può essere personalizzato per servire un numero di aziende verticali.

- **SAP Analytics Cloud**: soluzione per la visualizzazione dei dati per aziende di tutte le dimensioni. Offre pianificazione aziendale, analisi predittiva e funzionalità di reporting all'interno della suite. Permette agli utenti di compilare dati da diverse sorgenti. Le funzionalità per l'analisi dei dati comprendono scoperta dei dati, report ad hoc, pianificazione e previsione, analisi predittive e monitoraggio di KPI.

I valori chiave e i punti di forza di Knowage rispetto agli altri BI software sono diversi. Come prima cosa occorre sottolineare il fatto che sia open source, garantisce quindi libero accesso al codice sorgente favorendo la collaborazione con comunità internazionali. La suite viene sviluppata da professionisti e sono disponibili svariati servizi a supporto dei clienti per raggiungere con successo gli obiettivi di questi ultimi. Un altro grande vantaggio è il fatto che sia composto da svariati moduli, tutti concepiti per uno specifico dominio analitico e integrabili tra di loro per costruire una soluzione su misura. Infine, Knowage conferisce la capacità di auto gestirsi, in modo che l'utente finale possa costruire analisi private, esplorare e organizzare uno spazio di dati personalizzato.

# Capitolo 4

# Case study: progetto Pagoda

Il progetto Pagoda nasce con l'obiettivo di gestire i dati raccolti dall'area di Ingegneria Clinica dell'AUSL di Modena – Laboratorio Unificato di Baggiovara. I 10 milioni di esami annui dell'AUSL sono al servizio di 8 ospedali provinciali, 37 punti di prelievo sul territorio, 5 laboratori analisi territoriali interconnessi, tra i quali quello ad alta automazione di Baggiovara, tra i più grandi d'Europa. Il sistema informativo di Laboratorio gestisce ogni anno circa 2 milioni di richieste, 4 mln di campioni, 14 mln di prestazioni, 158 mln di analisi, 476 mln di risultati per un totale di 5 miliardi di dati a partire dal 2005. Nel laboratorio centralizzato dell'Ospedale di Baggiovara convergono tutti i campioni prelevati nelle decine di punti e strutture sul territorio, dove sono immediatamente identificati, tramite check-in via barcode o RFID (identificazione a radiofrequenza), e inseriti nel processo che si occupa di effettuare in maniera automatica tutte le analisi richieste. Ultimate le analisi, i referti sono firmati elettronicamente e immediatamente disponibili per il medico di reparto o di base attraverso il repository o il Fascicolo Elettronico regionale.

Questa mole di dati che descrive un sistema così complesso, ha reso necessario trasformare i dati stessi in informazioni che siano leggibili, interpretabili e gestibili dai diversi attori interessati. Tutto questo deve però essere fatto eliminando il rumore di fondo che deriva dalle interpretazioni dei vari attori

che mirano a "beneficiare" in modo diverso dei dati.

Nasce così Pagoda, la dashboard web di Engineering che consente di raccogliere e analizzare i dati generati dalle attività di uno o più laboratori di analisi, facilitandone l'esplorazione e la trasformazione in informazioni indispensabili per l'attività di controllo e ottimizzazione. Questo grazie anche a un vertical database che consente di utilizzare un ridotto numero di data mart e un'estrema fluidità nella fruizione del dato, superando i tradizionali datawarehouse con elevate performance nelle interrogazioni.

La piattaforma consente un monitoraggio in tempo reale della produttività e dei processi, statistiche con logica up down, controlli economici dei rendimenti delle tecnologie e dei consumi, appropriatezza predittiva e di follow-up. Pagoda permette di aggregare e rendere comprensibile e leggibile questa mole di dati sia in forma grafica che tabellare, con una profondità di analisi che va dal macrodato dipartimentale alla singola attività analitica, offrendo diversi tipi di interrogazioni con confronto su vari archi temporali. Vi è inoltre la possibilità di importare dati anche esterni al laboratorio, ad esempio da altre banche dati o fonti, consentendo di estendere le diverse possibilità di analisi. Pagoda consente ad ogni utente abilitato di creare il proprio cruscotto "smart", in modo che possa visualizzare le analisi, i report e gli indicatori di suo maggiore interesse, con la possibilità di esportare i dati nei formati più diffusi.

Grazie a Pagoda il manager di laboratorio può tenere sotto controllo in modo autonomo:

- **volumi**, grazie all'analisi dei dati relativi al numero di campioni, alla distribuzione oraria delle richieste, alla tipologia di esami, settori o unità operative richiedenti;

- **attività**, controllando in tempo reale l'intero processo di lavorazione attraverso il tracciamento dei singoli campioni segnalando poi eventuali anomalie (es. provette smarrite) o criticità operative (es. rallentamenti dei flussi di lavorazione);

- **risultati**, tramite la rappresentazione grafica della distribuzione dei risultati di un singolo test al fine di valutarne l'andamento rispetto ai valori di riferimento configurati;

- **qualità**, attraverso i dati delle "non conformità" e degli altri indicatori di qualità, come ad esempio l'andamento del TAT (turn around time) e la percentuale di sforamento temporale dei parametri dichiarati o di altri parametri personalizzabili;

- **appropriatezza**, tramite una previsione sull'impatto economico di una nuova regola di appropriatezza prima che questa venga effettivamente applicata o analizzando la percentuale di risparmio delle regole già utilizzate;

- **epidemiologia**, consentendo l'elaborazione in autonomia delle statistiche microbiologiche relative alla distribuzione di determinati ceppi o casistiche.

Le analisi svolte coinvolgono tutti i laboratori controllati dalla AUSL di Modena, denominati nel loro insieme come dipartimento, e le unità operative di cui ciascuno è composto. L'interesse si focalizza sulle richieste e sui vari elementi a esse collegate. Un paziente è collegato a una o più richiesta mentre una richiesta, o meglio un codice richiesta, è collegata univocamente a un solo paziente. Ogni richiesta può contenere una o più prestazioni alle quali possono essere collegate una o più analisi, che a loro volta sono collegate ai campioni, dove per campione si intende genericamente la provetta. Il risultato del campione può essere parte di uno o più referti.

La fase di accettazione di una richiesta, collegata con la prestazione, consiste nella stampa di un'etichetta che viene poi posizionata sul campione ed utilizzata durante la fase di check-in. Una richiesta è accettata quando viene inserita a sistema dal medico. Per quanto riguarda invece la fase di check-in, una richiesta è checkinata quando almeno un campione al suo interno è stato checkinato. Una richiesta, una volta accettata in un determinato laboratorio, può inviare le varie analisi in più di una UOP (unità operativa di produzione)

e anche a UOP non appartenenti al proprio dipartimento. Il calcolo della numerosità delle richieste è sempre dipendente dal livello considerato: in caso di analisi a livello di unità operativa, una stessa richiesta viene contata due volte se una sua analisi viene fatta in due UOP diverse; in caso invece di analisi a livello di laboratorio, la richiesta verrà contata una volta sola. Di conseguenza, se si contano tutte le richieste checkinate nelle varie UOP di un dato laboratorio, queste non coincideranno con le richieste checkinate a livello di quel laboratorio. L'ultima fase è quella di refertazione, che consiste nella produzione del referto finale che verrà poi consegnato al paziente.

Per quanto riguarda le caratteristiche più tecniche del progetto, quest'ultimo è stato realizzato utilizzando il software SpagoBI, versione antecedente del software Knowage. Vi sono due principali DataSource da cui vengono presi i dati per costruire i documenti di Pagoda:

- *PagodaVertica*, che punta su Vertica ed è il DataSource principale tramite cui sono stati fatti la maggior parte dei documenti;
- *Pagoda_RealTime*, che punta su Oracle ed è il DataSource realtime utilizzato per realizzare i documenti riguardanti le analisi in tempo reale.

Viene poi ancora utilizzato un terzo DataSource, che prende il nome di VERTICA PER MODELLI, che utilizza un dialetto diverso da *PagodaVertica* e viene principalmente utilizzato per la costruzione dei Qbe.

## 4.1 Porting di cruscotti e documenti OLAP

Engineering si pone come traguardo futuro quello di importare l'intero progetto Pagoda su Knowage. Nella prima sezione di questo lavoro si è voluto focalizzare l'attenzione su una parte di questo grande progetto e in particolare sulla voce di menù "Statistiche richieste". L'obiettivo è quindi quello di migrare i documenti di questa sezione da SpagoBI a Knowage cercando di ottimizzarli, utilizzando ad esempio per uno stesso documento un solo data

set anziché quattro, e di migliorarli, mettendo in risalto le qualità di Knowage rispetto a SpagoBI. Questa sezione di Pagoda si occupa di mostrare, da un punto di vista sia grafico che tabellare, i risultati relativi alle richieste erogate su diversi livelli di dettaglio. Comprende in totale sette documenti, di cui quattro sono di tipo reportistico e tre sono dei documenti OLAP. È stata utilizzata una versione di Knowage in locale in quanto non è stato ancora sviluppando un ambiente di test apposito per lavorare su Pagoda e a cui possono accedere gli utenti interessati.

Come prima cosa sono stati creati i DataSource da cui vengono reperiti i dati per poi passare alla creazione dei vari documenti composti.

### 4.1.1 Documento 1: "Richieste per laboratorio di accettazione"

Questo primo documento è un cruscotto dove vengono visualizzate le quantità di richieste accettate, checkinate e refertate a livello di dipartimento e di singolo laboratorio di accettazione rispetto alla data selezionata.

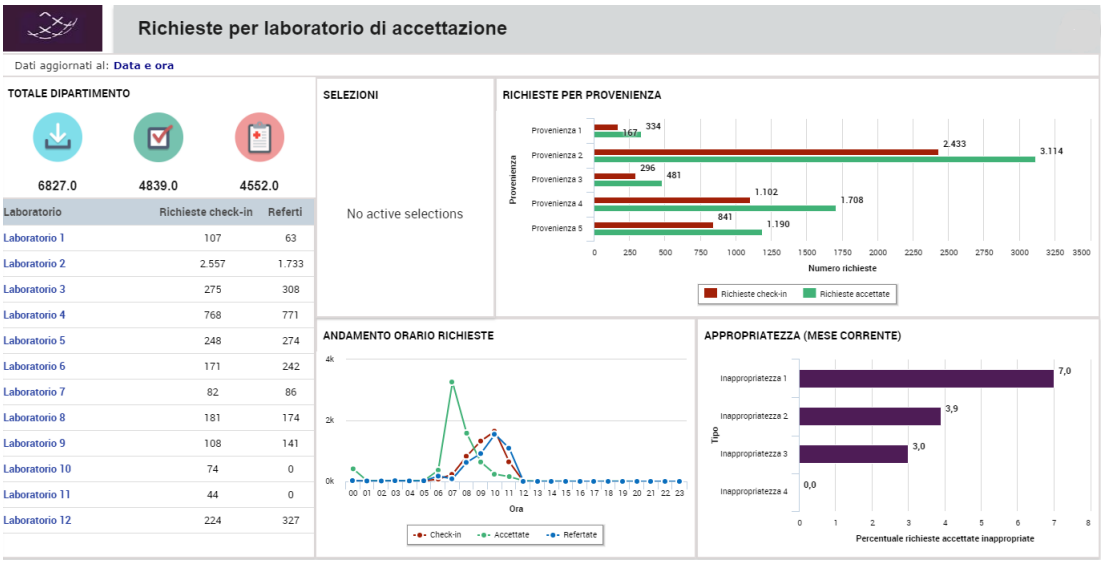


Figura 4.1: Cockpit "Richieste per laboratorio di accettazione"

Le misure prese in considerazione sono pertanto:

- **Richieste accettate**: una richiesta viene considerata accettata su un laboratorio se il reparto che effettua la richiesta appartiene a quel laboratorio. Esistono alcuni laboratori che non hanno dei reparti associati e di conseguenza questa misura può anche assumere valore nullo. La misura è additiva e quindi la somma delle richieste accettate per ogni laboratorio corrisponde alle richieste accettate per l'intero dipartimento.

- **Richieste check-in**: una richiesta è chekinata su un laboratorio se esiste almeno un campione associato a quella richiesta che ha fatto check-in su quel laboratorio. È possibile ci siano delle richieste con campioni che vengono analizzati da due distinti laboratori, in questo caso si avrà l'incremento del valore "richieste check-in" su entrambi i laboratori. La misura non è quindi additiva e la somma delle richieste accettate per ogni laboratorio non corrisponde alle richieste accettate per l'intero dipartimento.

- **Referti**: la misura mostra la quantità di referti prodotti all'interno delle UOP che compongono il laboratorio. Nel conteggio vengono considerati tutti i referti, sia quelli prodotti dai Sarf firmati che quelli non firmati (notturni). La misura è additiva e quindi la somma dei referti per ogni laboratorio corrisponde ai referti per dipartimento.

Per questo documento vengono utilizzati due diversi data set. Il primo viene usato per la rappresentazione dei dati in tempo reale e la query per la creazione del data set fa infatti riferimento al DataSource *Pagoda_ RealTime*. Grazie a questa interrogazione vengono estratti i dati relativi alle misure sopra elencate per un dato giorno e suddivisi per ora, provenienza e laboratorio. Il giorno è un filtro che l'utente può selezionare all'interno del documento scegliendo tra gli ultimi otto giorni a partire dalla data corrente. In caso non selezioni nessuna data viene inserito come parametro di default il giorno corrente. Poiché questi sono dati acquisiti in tempo reale, in cima al cruscotto è

possibile anche leggere l'ultima data e ora in cui questi sono stati aggiornati. Il secondo data set estrae invece i dati relativi alle inappropriatezze delle richieste accettate per un dato mese suddivisi per regole di appropriatezza e laboratorio e interroga il DataSource *PagodaVertica*. Per quanto riguarda il mese considerato per estrarre i dati, viene preso in considerazione quello corrispondente alla data selezionata dall'utente.

Si passa ora a illustrare nel dettaglio i vari grafici presenti all'interno del documento.


Figura 4.2: Grafico "Totale dipartimento"

| Laboratorio | Richieste check-in | Referti |
|---|---|---|
| Laboratorio 1 | 107 | 63 |
| Laboratorio 2 | 2.557 | 1.733 |
| Laboratorio 3 | 275 | 308 |
| Laboratorio 4 | 768 | 771 |
| Laboratorio 5 | 248 | 274 |
| Laboratorio 6 | 171 | 242 |
| Laboratorio 7 | 82 | 86 |
| Laboratorio 8 | 181 | 174 |
| Laboratorio 9 | 108 | 141 |
| Laboratorio 10 | 74 | 0 |
| Laboratorio 11 | 44 | 0 |
| Laboratorio 12 | 224 | 327 |

Figura 4.3: Tabella "Totali laboratori"

Nella Figura 4.2 vengono illustrate le quantità totali di richieste accettate, richieste check-in e referti per l'intero dipartimento.

La tabella di Figura 4.3 mostra le quantità totali di richieste check-in e referti per ogni laboratorio del dipartimento. Su questo widget è possibile selezionare, cliccando sul nome, un determinato laboratorio e gli altri grafici del documento si aggiorneranno di conseguenza mostrando i dati per quel particolare laboratorio.

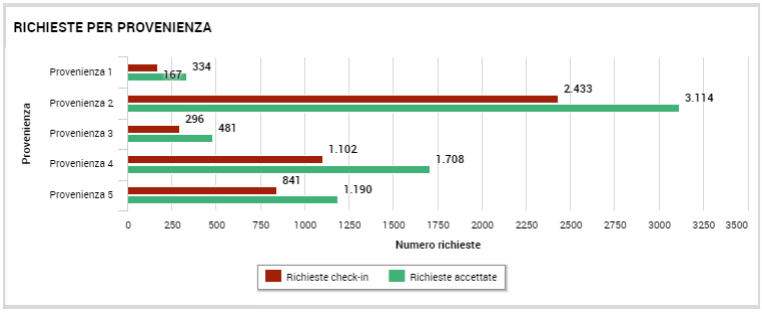


Figura 4.4: Grafico "Richieste per provenienza"

La Figura 4.4 mostra un grafico a barre con i valori del numero di richieste check-in e di richieste accettate per ogni origine. I dati mostrati all'apertura del report sono riferiti all'intero dipartimento, tuttavia è possibile filtrare per laboratorio tramite la Figura 4.3 e visualizzare le misure per un singolo laboratorio. È inoltre possibile selezionare all'interno di questo grafico una particolare provenienza e gli altri widget si modificheranno visualizzando le misure per quella particolare origine.

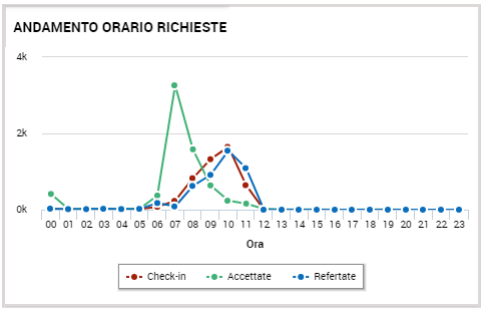


Figura 4.5: Grafico "Andamento orario richieste"

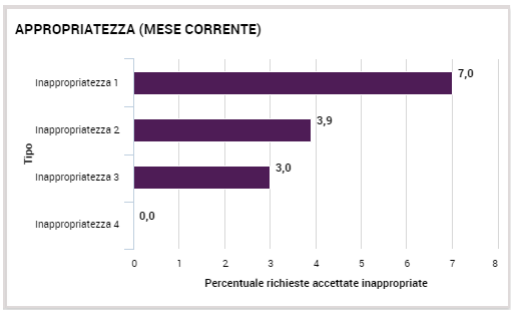


Figura 4.6: Grafico "Appropriatezza (mese corrente)"

Nella Figura 4.5 viene illustrato un grafico a linea che mostra l'andamento orario delle richieste accettate e ceckinate e dei referti. Anche in questo caso, poiché in origine le misure si riferiscono al dipartimento, è possibile vedere i dati per singolo laboratorio o per una particolare origine filtrando dalla

Figura 4.3 e dalla Figura 4.4. È anche possibile selezionare una determinata ora aggiornando di conseguenza gli altri widget.

La Figura 4.6 si focalizza invece solo sulle richieste accettate e in particolare su quelle inappropriate, ovvero che non dovevano essere accettate. Il grafico mostra la distribuzione percentuale delle tipologie di inappropriatezza sul mese corrente. Questo widget si aggiorna allo stesso modo degli altri ma non può modificare in alcun modo gli altri grafici.

Le selezioni fatte sui grafici abilitati vengono poi visualizzate su un apposito widget di selezione ed è possibile cancellarle in qualsiasi momento tornando quindi alla schermata iniziale con i dati riferiti all'intero dipartimento.

In una seconda pagina dello stesso documento sono state create delle tabelle di dettaglio per la visualizzazione e la lettura dei dati.

**DETTAGLIO PER LABORATORIO E PROVENIENZA**

| | Provenienza | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Provenienza 1 | | | Provenienza 2 | | | Provenienza 3 | | | Provenienza 4 | | | Provenienza 5 | | |
| Laboratorio | Accettate | Check-in | Refertate | Accettate | Check-in | Refertate | Accettate | Check-in | Refertate | Accettate | Check-in | Refertate | Accettate | Check-in | Refertate |
| Laboratorio 1 | 959.00 | 707.00 | 358.00 | 1,294.00 | 919.00 | 510.00 | 412.00 | 121.00 | 125.00 | 941.00 | 679.00 | 619.00 | 251.00 | 131.00 | 121.00 |
| Laboratorio 2 | | | | 0.00 | 3.00 | 0.00 | 0.00 | 4.00 | 4.00 | 0.00 | 100.00 | 59.00 | | | |
| Laboratorio 3 | 60.00 | 33.00 | 26.00 | 10.00 | 5.00 | 4.00 | 9.00 | 9.00 | 8.00 | 313.00 | 206.00 | 268.00 | 63.00 | 22.00 | 2.00 |
| Laboratorio 4 | 0.00 | 0.00 | 14.00 | 41.00 | 41.00 | 59.00 | 24.00 | 6.00 | 7.00 | 1,506.00 | 721.00 | 691.00 | | | |
| Laboratorio 5 | 0.00 | 54.00 | 18.00 | 0.00 | 30.00 | 18.00 | 0.00 | 6.00 | 10.00 | 0.00 | 158.00 | 228.00 | | | |
| Laboratorio 6 | | | | | | | 0.00 | 2.00 | 6.00 | 0.00 | 169.00 | 236.00 | | | |
| Laboratorio 7 | | | | 0.00 | 1.00 | 1.00 | | | | 121.00 | 81.00 | 85.00 | | | |
| Laboratorio 8 | | | | | | | | | | 139.00 | 108.00 | 141.00 | | | |
| Laboratorio 9 | 25.00 | 20.00 | 45.00 | 75.00 | 70.00 | 50.00 | 16.00 | 12.00 | 8.00 | 85.00 | 65.00 | 71.00 | 18.00 | 14.00 | 0.00 |
| Laboratorio 10 | | | | 0.00 | 3.00 | 0.00 | 0.00 | 6.00 | 0.00 | 0.00 | 65.00 | 0.00 | | | |
| Laboratorio 11 | | | | | | | | | | 0.00 | 44.00 | 0.00 | | | |
| Laboratorio 12 | 0.00 | 27.00 | 77.00 | 0.00 | 30.00 | 199.00 | 0.00 | 130.00 | 9.00 | 0.00 | 37.00 | 42.00 | | | |
| Total | 1,044.00 | 841.00 | 538.00 | 1,420.00 | 1,102.00 | 841.00 | 461.00 | 296.00 | 177.00 | 3,105.00 | 2,433.00 | 2,440.00 | 332.00 | 167.00 | 123.00 |

**DETTAGLIO PER LABORATORIO E ORA**

| | 00 | | | 01 | | | 02 | | | 03 | | | 04 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Laboratorio | Accettate | Check-in | Refertate | Accettate | Check-in | Refertate | Accettate | Check-in | Refertate | Accettate | Check-in | Refertate | Accettate | Check-in | Refertate | Accettat |
| Laboratorio 1 | 274.00 | 5.00 | 6.00 | 3.00 | 2.00 | 4.00 | 2.00 | 1.00 | 1.00 | 6.00 | 5.00 | 2.00 | 9.00 | 5.00 | 6.00 | 5.00 |
| Laboratorio 2 | | | | | | | | | | | | | | | | |
| Laboratorio 3 | 72.00 | 3.00 | 6.00 | 2.00 | 1.00 | 4.00 | 10.00 | 5.00 | 2.00 | 2.00 | 5.00 | 7.00 | 2.00 | 2.00 | 1.00 | 1.00 |
| Laboratorio 4 | 42.00 | 7.00 | 10.00 | 7.00 | 4.00 | 5.00 | 10.00 | 3.00 | 6.00 | 3.00 | 7.00 | 10.00 | 5.00 | 1.00 | 1.00 | 16.00 |
| Laboratorio 5 | | | | | | | | | | | | | | | | |
| Laboratorio 6 | | | | | | | | | | | | | | | | |
| Laboratorio 7 | 2.00 | 0.00 | 0.00 | | | | 1.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | | | | 1.00 |
| | | | 5.00 | 2.00 | 2.00 | 2.00 | 3.00 | 5.00 | 5.00 | 3.00 | 1.00 | 5.00 | 8.00 | 5.00 | 8.00 | 4.00 |
| CHARTS | CROSS TABLE | | 0.00 | | | | | | | | | | | | | |

Figura 4.7: Tabelle di dettaglio

Nella prima tabella vengono visualizzati i valori delle tre misure per ogni laboratorio e origine e al fondo di ogni riga e di ogni colonna sono calcolati i totali per un determinato laboratorio e per una particolare provenienza. Nella seconda vengono sempre raffigurati i valori delle misure di interesse ma

per ogni laboratorio e fascia oraria e i totali vengono calcolati al termine di ogni riga e colonna.

### 4.1.2 Documento 2: "Richieste per laboratorio di produzione"

Questo documento è un cruscotto in cui vengono visualizzate le quantità di richieste checkinate e refertate rispetto alla data e al laboratorio di produzione selezionati.

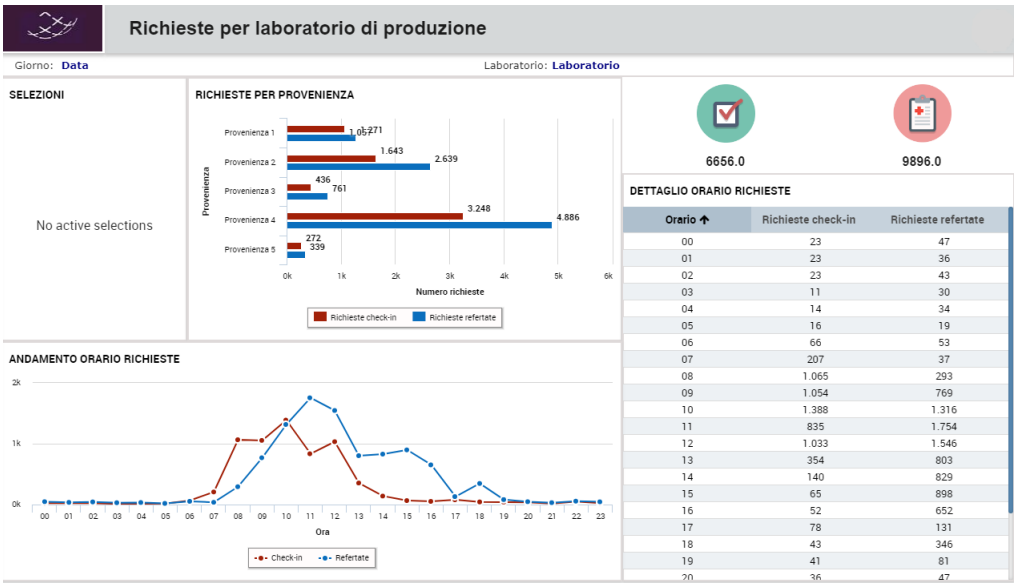


Figura 4.8: Cockpit "Richieste per laboratorio di produzione"

Le misure prese in considerazione sono analoghe a quelle analizzate nel primo documento con la differenza che in questo caso si riferiscono ai laboratori di produzione:

- **Richieste check-in**: una richiesta è chekinata su un laboratorio se esiste almeno un campione associato a quella richiesta che ha fatto check-in su quel laboratorio. La misura non è additiva.

- **Richieste refertate**: una richiesta è refertata per dipartimento quando tutte le prestazioni sono comparse almeno una volta su un referto, sia esso firmato o non firmato. Una richiesta è invece refertata per un singolo laboratorio quando tutte le prestazioni che devono essere eseguite su quel particolare laboratorio sono comparse almeno una volta su un referto, sia esso firmato o non firmato. Questa misura non è additiva, ovvero la somma delle richieste refertate per ogni laboratorio non corrisponde alle richieste refertate per dipartimento.

Per la creazione dei grafici di questo documento viene utilizzato un solo data set e l'interrogazione punta al DataSource *PagodaVertica*. Tramite quest'ultima vengono estratti i dati relativi alle misure appena citate per una determinata data e un dato laboratorio e suddivisi per origine della richiesta e orario. Il giorno e il laboratorio sono due parametri di ingresso che possono essere selezionati direttamente dall'utente all'apertura del documento. Come data si può scegliere tra una qualsiasi data del calendario evitando però di selezionare il giorno corrente, a causa dei dati non ancora caricati, ed è obbligatorio effettuare una scelta in quanto non ha nessun parametro di default. Il laboratorio, invece, è possibile selezionarlo dalla lista dei laboratori di produzione oppure, se non si vogliono visualizzare i dati per un laboratorio specifico, si può anche selezionare l'intero dipartimento. In quest'ultimo caso l'utente può anche non effettuare una scelta e viene preso come parametro di default l'intero dipartimento anziché un laboratorio specifico. In cima al cockpit è inoltre possibile visualizzare i parametri selezionati per la visualizzazione dei dati.

Nella Figura 4.9 vengono mostrate le quantià totali delle richieste check-in e delle richieste refertate alla data e al laboratorio selezionati. La Figura 4.10 è un grafico a barre che illustra l'andamento delle due misure in base all'origine. È possibile cliccare su questo widget e selezionare una particolare provenienza in modo che gli altri grafici si aggiornino e mostrino i dati solo per quella provenienza.

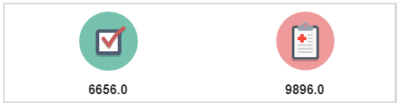


Figura 4.9: Grafico "Totale laboratorio"

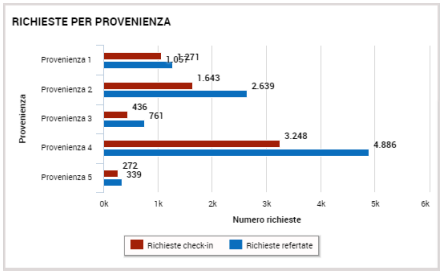


Figura 4.10: Grafico "Richieste per provenienza"

Le immagini sotto riportate si focalizzano sui dati delle richieste suddivise per fascia oraria. La Figura 4.11 è un grafico a linea che mostra l'andamento orario delle due misure, mentre la Figura 4.12 è una tabella di dettaglio con la distribuzione oraria delle misure. Al fondo della tabella si leggono inoltre i totali delle richieste check-in e delle richieste refertate. Anche in questo caso, come nella Figura 4.10, è possibile filtrare per orario selezionandolo dalla tabella di dettaglio e aggiornare in questo modo gli altri widget del documento.

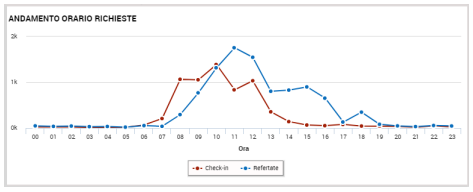


Figura 4.11: Grafico "Andamento orario richieste"

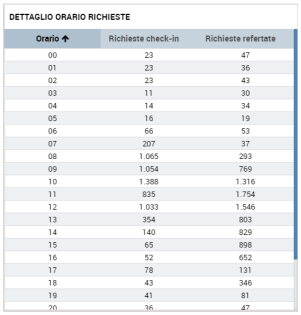


Figura 4.12: Tabella "Dettaglio orario richieste"

I filtri applicati cliccando sui vari grafici sono visibili all'interno di un widget di selezione, quest'ultimo consente di eliminare le selezioni nel caso in cui si volesse tornare a visualizzare i dati nella forma in cui erano all'apertura del cruscotto.

### 4.1.3 Documento 3: "Richieste per unità operativa di produzione"

Questo cruscotto è analogo a quello del paragrafo precedente con la differenza che in questo cockpit vengono visualizzati i dati per una particolare unità operativa di un dato laboratorio di produzione.

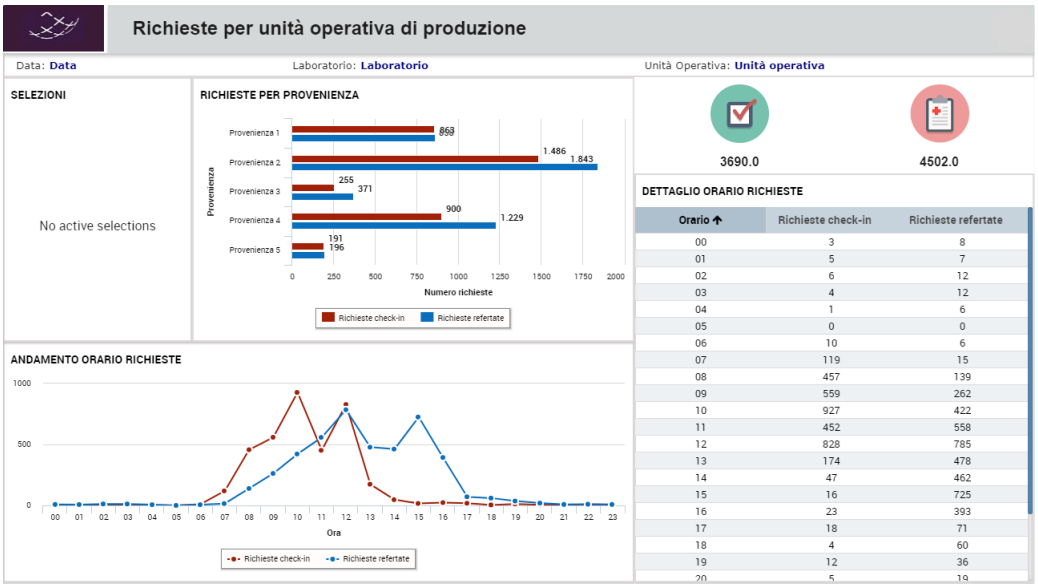


Figura 4.13: Cockpit "Richieste per unità operativa di produzione"

Le misure analizzate in questo documento sono:

- **Richieste check-in**: una richiesta è chekinata su un laboratorio se esiste almeno un campione associato a quella richiesta che ha fatto check-in su quel laboratorio. La misura non è additiva.
- **Richieste refertate**: una richiesta è refertata per dipartimento quando tutte le prestazioni sono comparse almeno una volta su un referto. Una richiesta è refertata per un singolo laboratorio quando tutte le prestazioni che devono essere eseguite su quel particolare laboratorio sono comparse almeno una volta su un referto. Infine, una richiesta è refertata per una determinata UOP quando tutte le prestazione che

devono essere eseguite su quella unità operativa sono comparse almeno una volta su un referto, sia esso firmato o non firmato. Questa misura non è additiva, ovvero la somma delle richieste refertate per UOP non corrisponde alle richieste refertate per laboratorio afferente delle UOP.

I dati fanno riferimento a un solo data set che punta al DataSource *PagodaVertica*. Vengono estratti tramite una query che seleziona le misure di interesse per una certa data, un particolare laboratorio e un'unità operativa afferente al laboratorio selezionato e sono suddivisi per fascia oraria e provenienza. All'interno del documento ci sono tre parametri di ingresso: ognuno di essi deve essere selezionato obbligatoriamente dall'utente perché non sono stati definiti dei valori di default. Il primo parametro è la data e può essere selezionata tra quelle del calendario con l'unica limitazione di non selezionare il giorno corrente. Un secondo filtro è il laboratorio di produzione che si può scegliere da una finestra popup tra quelli esistenti; in questo documento non è possibile selezionare l'intero dipartimento. L'ultimo parametro è l'unità operativa di produzione che viene selezionata da una lista contenente tutte le UOP afferenti al laboratorio scelto in precedenza. In cima al cruscotto si possono leggere i parametri di ingresso selezionati.

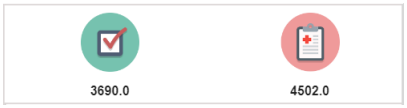


Figura 4.14: Grafico "Totale unità operativa"

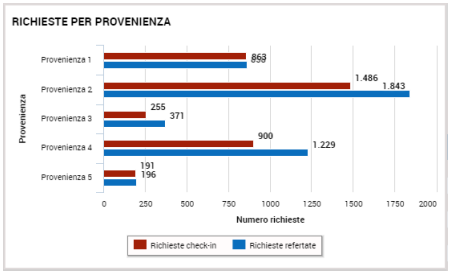


Figura 4.15: Grafico "Richieste per provenienza"

La Figura 4.14 e la Figura 4.15 mostrano rispettivamente le quantità totali e il numero di richieste per origine delle misure di interesse del documento. Nel grafico di Figura 4.15 è possibile filtrare per provenienza e aggiornare di conseguenza gli altri widget.

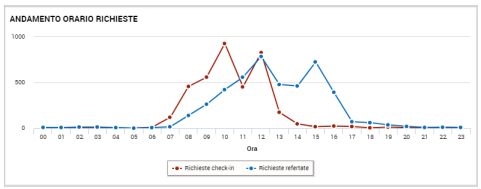


Figura 4.16: Grafico "Andamento orario richieste"

DETTAGLIO ORARIO RICHIESTE

| Orario | Richieste check-in | Richieste refertate |
|---|---|---|
| 00 | 3 | 8 |
| 01 | 5 | 7 |
| 02 | 6 | 12 |
| 03 | 4 | 12 |
| 04 | 1 | 6 |
| 05 | 0 | 0 |
| 06 | 10 | 6 |
| 07 | 119 | 15 |
| 08 | 457 | 139 |
| 09 | 559 | 262 |
| 10 | 927 | 422 |
| 11 | 452 | 558 |
| 12 | 828 | 785 |
| 13 | 174 | 478 |
| 14 | 47 | 462 |
| 15 | 16 | 725 |
| 16 | 23 | 393 |
| 17 | 18 | 71 |
| 18 | 4 | 60 |
| 19 | 12 | 36 |
| 20 | 5 | 19 |

Figura 4.17: Tabella "Dettaglio orario richieste"

La Figura 4.16 e la Figura 4.17 trattano i dati suddivisi per fascia oraria. Nel primo le misure vengono visualizzate tramite un grafico a linea, nel secondo viene invece utilizzata una tabella di dettaglio dove è possibile leggere i dati e i totali per ogni misura e suddivisi per fascia oraria. Tramite la tabella di dettaglio è possibile filtrare per orario e i filtri selezionati verranno visualizzati nel widget di selezione.

### 4.1.4 Documento 4: "Richieste per prestazione"

Il documento è l'ultimo dei cruscotti trattati all'interno di questa sezione di Pagoda e si focalizza sulle richieste erogate e loro valorizzazione rispetto ai parametri di ingresso selezionati.

Le misure prese in considerazione sono pertanto:

- **Richieste refertate**: si distingue tra richieste refertate per dipartimento, per laboratorio e per UOP. La misura non è additiva, ovvero la somma delle richieste refertate per unità operativa non corrisponde alle richieste refertate per laboratorio afferente delle UOP e la somma delle richieste refertate per laboratorio non corrisponde alle richieste refertate per dipartimento. Si dice che un referto è in ritardo quando viene prodotto dopo la data prevista, dove con data prevista si intende la massima data prevista per l'erogazione delle prestazioni conte-

nute nel referto. È possibile quindi avere delle prestazioni in ritardo appartenenti a referti non in ritardo.

- **Prestazione erogata**: la data di erogazione corrisponde alla prima volta in cui la prestazione compare su un referto, sia esso firmato o non firmato. Una prestazione risulta in ritardo quando viene erogata dopo la data prevista di erogazione. La data prevista di erogazione dipende dai giorni di lavorazione configurati su Openlis.

- **Valorizzazione economica**: alla data in cui una richiesta viene erogata viene attribuita la quota economica delle prestazioni che compongono la richiesta utilizzando come valorizzazione il tariffario SSN. Poiché la misura "Richieste refertate" non è additiva per dipartimento, laboratorio e UOP, anche questa misura non è additiva.

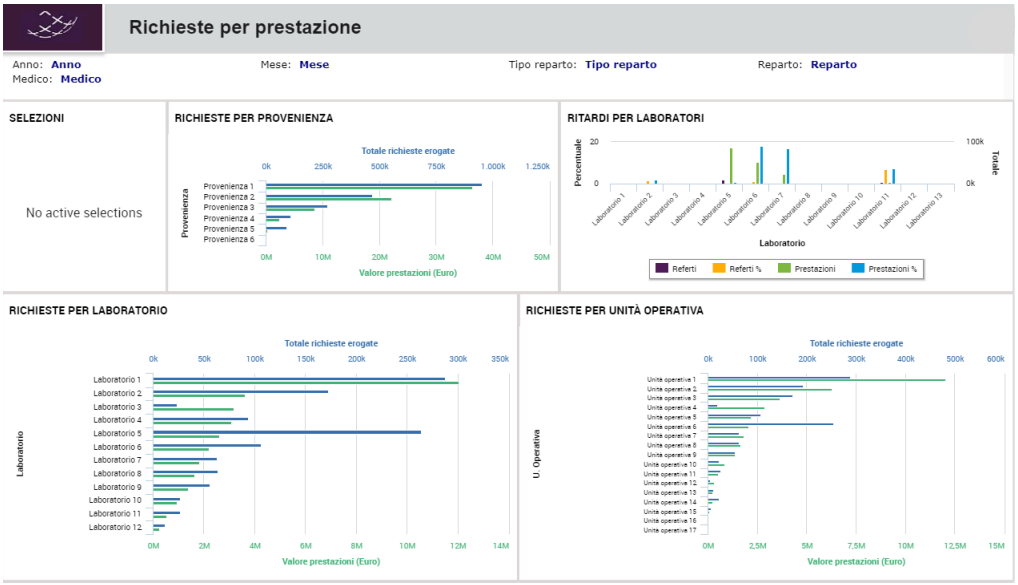


Figura 4.18: Cockpit "Richieste per prestazione"

Per questo cruscotto è stato necessario creare quattro data set diversi per ogni grafico e tutti puntano al DataSource *PagodaVertica*. Per quanto riguarda i data set relativi ai tre grafici a barre orizzontali, le query sono molto simili,

fatta eccezione per il fatto che puntano a tabelle di dati diverse. Le interrogazioni estraggono i dati relativi alle richieste erogate e alle valorizzazioni per un determinato anno suddivisi per provenienza in un caso, per laboratorio in un altro e per UOP nell'ultimo grafico. È inoltre possibile filtrare i dati per mese, medico, reparto e tipo di reparto ma sono dei parametri facoltativi all'interno della query e per questo motivo sono stati inseriti tramite uno script in Javascript. Riporto di seguito lo script relativo a questi parametri.

```
var var_mese ="";
var var_medico = "";
var var_reparto = "";
var var_grupporeparto = "";

if (parameters.get('mese')!=null && parameters.get('mese')!="'0'") {
  var_mese = "AND to_number(MESE_REFERTAZIONE) IN ($P{mese})";
}

if (parameters.get('medico')!=null && parameters.get('medico')!=0) {
  var_medico= "AND CD_MEDICO  IN ($P{medico})";
}

if (parameters.get('reparto')!=null && parameters.get('reparto')!=0) {
  var_reparto= "AND CD_REPARTO IN ($P{reparto})";
}

if (parameters.get('gruppo_reparto')!=null &&
    parameters.get('gruppo_reparto')!=0) {
  var_grupporeparto= "AND CD_TIPOREPARTO IN ($P{gruppo_reparto})";
}

query = query.replace("PLACEHOLDER_MESE", var_mese);
query = query.replace("PLACEHOLDER_MEDICO", var_medico);
```

```
query = query.replace("PLACEHOLDER_REPARTO", var_reparto);
query = query.replace("PLACEHOLDER_GRUPPOREPARTO", var_grupporeparto);
```

Questo permette di specificare cosa inserire nella query nel caso in cui l'utente decida di selezionare dei valori oppure non selezioni nulla. Viene definita una variabile per ogni clausola nel `where` che si vuole sostituire. Tramite la condizione di `if` si verifica se il parametro considerato è associato a dei valori specificati dall'utente. In caso lo sia viene inserita un'opportuna condizione di `where`, sostituendo il `PLACEHOLDER_NOME-PARAMETRO` nella query con la variabile dello script.

L'ultimo data set calcola la percentuale e il numero di referti e prestazioni in ritardo per un particolare anno e suddivisi per laboratorio. Anche in questo caso è possibile filtrare l'interrogazione per i parametri sopra citati.

Il documento ha cinque parametri di ingresso. All'apertura del cruscotto viene richiesto di inserire l'anno in quanto è l'unico parametro a dover essere inserito obbligatoriamente dall'utente e questo viene scelto da una lista degli anni di cui si hanno i dati. I parametri restanti possono invece non essere associati a nessun valore e in questo caso non viene applicato il filtro e i dati visualizzati sono filtrati solo per anno. È a discrezione dell'utente scegliere se vuole vedere i dati per, ad esempio, un determinato mese e tipo di reparto e, ovviamente, può filtrare per uno solo o più di questi parametri. Il tipo di reparto viene scelto da una lista delle due opzioni disponibili: interno o esterno. I parametri mese, medico e reparto sono invece multivalore, per questi è quindi possibile filtrare per più di un valore. Il mese può essere selezionato da una lista contenente tutti i mesi, il medico è scelto invece da una finestra pop up contenente tutti i nomi dei medici e, infine, il reparto è selezionato anch'esso grazie a una finestra pop up che racchiude le descrizioni di ogni reparto. In cima al documento è possibile leggere i parametri scelti dall'utente; in caso non ci sia nessun filtro per mese, medico, tipo reparto e reparto si legge che sono stati scelti tutti i mesi, tutti i medici, tutti i tipi di reparto e tutti i reparti ed è quindi stata fatta una somma delle misure.

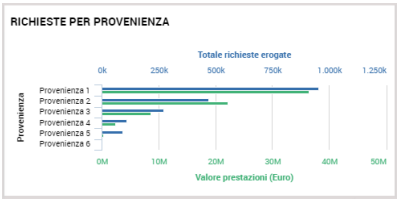


Figura 4.19: Grafico "Richieste per provenienza"

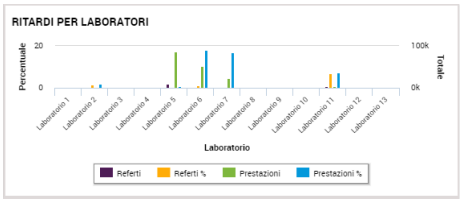


Figura 4.20: Grafico "Ritardi per laboratori"

Nel grafico a barre di Figura 4.19 viene mostrato il numero di richieste prenotate erogate per ogni provenienza e la loro valorizzazione, mentre in quello di Figura 4.20 si visualizza il numero di referti e prestazioni in ritardo e il loro valore percentuale suddivisi per laboratorio e dipartimento. In quest'ultimo widget, se si vogliono visualizzare solo alcune delle quattro misure analizzate, è possibile deselezionare le misure di non interesse tramite la legenda posta sotto il grafico. Selezionando poi un'origine nel grafico di Figura 4.19 vengono aggiornati i grafici delle figure 4.21 e 4.22 usando quell'origine come filtro.

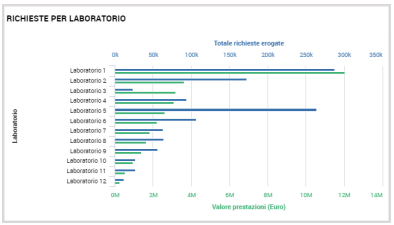


Figura 4.21: Grafico "Richieste per laboratorio"

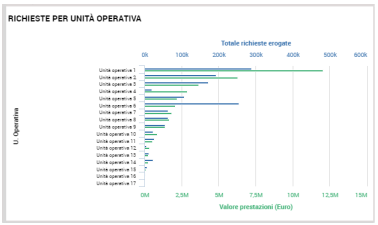


Figura 4.22: Grafico "Richieste per unità operativa"

I grafici di Figura 4.21 e di Figura 4.22 trattano le stesse misure, ovvero il numero di richieste erogate e loro valorizzazione, ma nel primo caso i dati sono per ogni laboratorio mentre nel secondo sono suddivisi per UOP. Selezionando un laboratorio nel grafico di Figura 4.21 viene aggiornato il grafico di Figura 4.22 utilizzando il laboratorio come filtro. Di conseguenza i dati del widget di Figura 4.22 possono essere filtrati sia per origine che per laboratorio grazie agli altri grafici presenti nel documento. Quest'ultimo widget

ha una particolarità rispetto a tutti quelli visti finora. Selezionando una particolare UOP è infatti possibile accedere a un secondo cruscotto che mostra le informazioni di dettaglio per quella particolare unità operativa. Questa procedura prende il nome di cross navigation.

**Dettaglio richieste per prestazione**

Anno: **Anno** Mese: **Mese** Tipo reparto: **Tipo reparto** Reparto: **Reparto**
Medico: **Medico** Provenienza: **Provenienza** Unita operativa: **Unità operativa**

| Codice | Prestazione | Valore | Totale erogate | Valore totale | Prestazioni in ritardo |
|---|---|---|---|---|---|
| Codice 1 | Prestazione 1 | € 2,00 | 7.307 | € 14.614,00 | 0 |
| Codice 2 | Prestazione 2 | € 4,00 | 11 | € 44,00 | 0 |
| Codice 3 | Prestazione 3 | € 4,00 | 61.365 | € 245.460,00 | 3 |
| Codice 4 | Prestazione 4 | € 6,00 | 1.953 | € 11.718,00 | 0 |
| Codice 5 | Prestazione 5 | € 6,00 | 8.407 | € 50.442,00 | 0 |
| Codice 6 | Prestazione 6 | € 15,75 | 9.189 | € 144.726,75 | 2 |
| Codice 7 | Prestazione 7 | € 10,00 | 905 | € 9.050,00 | 7 |
| Codice 8 | Prestazione 8 | € 3,00 | 124.814 | € 374.442,00 | 8 |
| Codice 9 | Prestazione 9 | € 3,00 | 110.583 | € 331.749,00 | 8 |
| Codice 10 | Prestazione 10 | € 3,00 | 295 | € 885,00 | 1 |
| Codice 11 | Prestazione 11 | € 3,00 | 62.340 | € 187.020,00 | 2 |
| Codice 12 | Prestazione 12 | € 3,00 | 11.990 | € 35.970,00 | 0 |
| Codice 13 | Prestazione 13 | € 15,70 | 177 | € 2.778,90 | 0 |
| Codice 14 | Prestazione 14 | € 15,70 | 227 | € 3.563,90 | 1 |
| Codice 15 | Prestazione 15 | € 15,70 | 83 | € 1.303,10 | 0 |
| Codice 16 | Prestazione 16 | € 15,70 | 749 | € 11.759,30 | 17 |
| Codice 17 | Prestazione 17 | € 15,70 | 239 | € 3.752,30 | 1 |
| Codice 18 | Prestazione 18 | € 15,70 | 47 | € 737,90 | 0 |
| Codice 19 | Prestazione 19 | € 15,70 | 202 | € 3.171,40 | 1 |
| Codice 20 | Prestazione 20 | € 15,70 | 159 | € 2.496,30 | 1 |
| Codice 21 | Prestazione 21 | € 15,70 | 31 | € 486,70 | 0 |
| Codice 22 | Prestazione 22 | € 15,70 | 366 | € 5.746,20 | 3 |
| Codice 23 | Prestazione 23 | € 15,70 | 365 | € 5.730,50 | 3 |

Page 1 1 - 23 of 219

Figura 4.23: Cockpit “Dettaglio richieste per prestazione”

Per la creazione del grafico sopra riportato è stato utilizzato un nuovo data set che punta sempre al DataSource *PagodaVertica*. Nel documento vengono riportati gli stessi filtri applicati nel cockpit iniziale ma è anche possibile modificarli solo per la tabella di dettaglio. I parametri impiegati vengono visualizzati sempre in cima al cruscotto e, in aggiunta a quelli del primo documento, si può leggere anche l’unità operativa dalla quale è stato caricato il documento. La tabella è costituita da una riga per ogni codice e nome di prestazione per quella particolare UOP. Per ogni prestazione si ha il valore, il numero di erogate, il valore totale (dato dalla moltiplicazione del totale di prestazioni erogate di quel tipo per il valore della prestazione) e quante prestazioni di quel genere sono in ritardo. In base all’unità operativa selezionata si avrà un numero diverso di righe e di pagine della tabella. È poi possibile ritornare al documento originale.

### 4.1.5 Documento 5: "Confronto richieste per laboratorio"

Il seguente è un documento OLAP che permette di analizzare in modo interattivo dati multidimensionali da prospettive multiple.


| | Totale Erogato | Valore | Prestazioni in ritardo | Referti in ritardo |
|---|---|---|---|---|
| ⊟ Gruppo 1 | 1,197,214 | EUR 33,392,931 | 86,023 | 9,775 |
| Laboratorio 1 | 264,159 | EUR 2,632,198 | 2,380 | 1,704 |
| Laboratorio 2 | 172,304 | EUR 3,633,873 | 51,596 | 2,753 |
| Laboratorio 3 | 94,117 | EUR 3,085,903 | 1,031 | 380 |
| Laboratorio 4 | 55,956 | EUR 1,387,063 | 26 | 23 |
| Laboratorio 5 | 106,808 | EUR 2,203,516 | 13 | 19 |
| Laboratorio 6 | 26,517 | EUR 541,397 | 5 | 5 |
| Laboratorio 7 | 288,036 | EUR 12,052,327 | 2,312 | 456 |
| Laboratorio 8 | 63,074 | EUR 1,843,013 | 22,803 | 774 |
| Laboratorio 9 | 26,461 | EUR 933,839 | 4,687 | 3,052 |
| Laboratorio 10 | 23,730 | EUR 3,189,097 | 948 | 563 |
| Laboratorio 11 | 12,000 | EUR 238,069 | 7 | 8 |
| Laboratorio 12 | 64,052 | EUR 1,652,636 | 215 | 38 |

Figura 4.24: OLAP "Confronto richieste per laboratorio"

Le misure qui mostrate sono:

- **Richieste erogate e loro valorizzazione**: in particolare il report mostra solo le richieste erogate per laboratorio alle quali viene associata, una volta erogate, una quota economica. Poiché la misura non è additiva per dipartimento, laboratorio e UOP, eliminando dal report il raggruppamento dei laboratori non si ottengono le richieste per dipartimento ed è per questo che ci si concentra solo sulle richieste erogate per laboratorio.
- **Prestazioni e referti in ritardo**: una prestazione o un referto risultano in ritardo quando vengono prodotti dopo la data prevista di erogazione.

Il DataSource di riferimento per prelevare i dati è sempre *PagodaVertica* e non vi è una query che estrae i dati ma è stata creata una tabella apposita su Vertica per l'estrazione delle informazioni di interesse. Le misure vengono visualizzate per un particolare anno che corrisponde quindi al parametro di ingresso del documento. All'apertura del documento viene richiesto all'utente di scegliere l'anno da considerare da una lista degli anni di refertazione disponibili, che verrà poi visualizzato in cima al documento. Non è settato alcun valore di default quindi l'utente è obbligato a selezionare un valore che è possibile poi modificare in qualsiasi momento.

La parte centrale del documento OLAP è una tabella pivot in cui, inizialmente, vengono mostrate le misure suddivise per laboratorio e su cui è possibile applicare svariate operazioni. Prima fra tutte è quella di poter modificare i filtri e i raggruppamenti del report utilizzando i parametri a disposizione: l'anno di erogazione, i laboratori, l'origine, il tipo di reparto (con i soli valori esterno o interno), il reparto e i medici. Trascinando infatti uno di questi riquadri su riga o colonna si modificano i raggruppamenti e si riescono a visualizzare i dati non solo più per laboratorio. Mentre, per i filtri, è possibile selezionare determinati valori tra i parametri disponibili nei pannelli sopra la tabella. I reparti sono tra i parametri a poter essere filtrati e sono suddivisi a due livelli: prima per radice del codice, ad esempio SBA o CA1, e poi per codice reparto. Se è di interesse per l'utente, è inoltre possibile visualizzare solo una parte delle misure presenti all'apertura del documento.

Esiste poi un menu apribile tramite l'icona posizionata a destra del documento e utilizzato per:

- scegliere tra diverse rappresentazioni di dati;
- scegliere tra diversi tipi di drill;
- applicare funzionalità che modificano la tabella pivot;
- ottenere dati addizionali basati sul modello caricato.


Menu documento OLAP

Il menu è suddiviso in tre sottosezioni: opzioni di drill, funzioni OLAP e funzioni della tabella. I diversi tipi di drill consistono in: position, member, replace e drill through. I primi tre sono riferiti alle dimensioni, mentre l'ultimo fa riferimento ai dati. L'utente può selezionare uno di questi tipi di drill cliccando sui bottoni appositi del menu.

Il drill position è il tipo di default e, quando selezionato, espanderà o collasserà, a seconda che si abbia un drill down o un drill up, la tabella pivot con i membri figli di una dimensione. Il drill member viene selezionato quando l'utente vuole eseguire l'operazione di drill non solo su un membro ma su tutti i membri dello stesso nome e livello. L'ultimo tipo di drill sulle dimensione è il drill replace che permette di sostituire un membro padre con i suoi membri figli. Infine, per eseguire il drill through, dopo aver cliccato sul bottone occorre selezionare una cella di cui si vogliono vedere i risultati. Si aprirà una finestra pop up dove l'utente può scegliere il livello di dettaglio con il quale i dati verranno visualizzati. Per fare questo occorre selezionare i livelli delle gerarchie di interesse e applicarli ai dati. L'utente può anche selezionare il numero massimo di righe da caricare ed esportare i dati caricati in formato csv.

Nella sezione delle funzioni OLAP è presente un solo pulsante che permette di visualizzare la query MDX corrente.

Passando infine alla sezione delle funzionalità, si hanno quattro bottoni. Il primo si chiama "Show parent members" e permette di vedere informazioni addizionali sulle dimensioni mostrate nella tabella pivot. Il secondo pulsante prende il nome di "Hide spans" e serve per nascondere o mostrare le estensioni. Altra funzione è quella definita "Show properties"; in un documento OLAP le proprietà dei membri XML, se configurate, possono essere rappre-

sentate in due modi possibili: come parte della tabella pivot, dove i valori di una proprietà sono posizionati sulle righe o sulle colonne oppure in una pop up come proprietà compatte. In questo caso è possibile solo la prima rappresentazione perché per la seconda non c'è il bottone. Infine, l'ultimo pulsante è "Suppress empty rows/columns" e consente di nascondere le righe e/o colonne vuote dalla tabella pivot.

### 4.1.6 Documento 6: "Confronto richieste per unità operativa"


| | Totale Erogato | Valore | Prestazioni in ritardo | Referti in ritardo |
|---|---|---|---|---|
| ⊖ Gruppo 1 | 1,315,003 | EUR 36,268,166 | 86,017 | 9,936 |
| Unità operativa 1 | 19,623 | EUR 2,860,121 | 557 | 378 |
| Unità operativa 2 | 4,674 | EUR 327,738 | 391 | 185 |
| Unità operativa 3 | 288,036 | EUR 12,052,327 | 2,312 | 456 |
| Unità operativa 4 | 22,126 | EUR 233,429 | 32 | 80 |
| Unità operativa 5 | 106,743 | EUR 2,185,507 | 13 | 17 |
| Unità operativa 6 | 1,704 | EUR 17,977 | | 2 |
| Unità operativa 7 | 253,915 | EUR 2,084,521 | 1,061 | 1,179 |
| Unità operativa 8 | 26,517 | EUR 541,397 | 5 | 5 |
| Unità operativa 9 | 55,940 | EUR 1,380,659 | 26 | 23 |
| Unità operativa 10 | 606 | EUR 6,393 | | |
| Unità operativa 11 | 172,304 | EUR 3,633,873 | 51,596 | 2,753 |
| Unità operativa 12 | 63,074 | EUR 1,843,013 | 22,803 | 774 |
| Unità operativa 13 | 64,052 | EUR 1,652,636 | 215 | 38 |
| Unità operativa 14 | 23,475 | EUR 840,209 | 4,408 | 2,558 |
| Unità operativa 15 | 7,080 | EUR 92,687 | 279 | 496 |
| Unità operativa 16 | 12,000 | EUR 238,069 | 7 | 8 |
| Unità operativa 17 | 193,134 | EUR 6,277,609 | 2,312 | 984 |

Figura 4.25: OLAP "Confronto richieste per unità operativa"

Le misure analizzate in questo documento OLAP sono le seguenti:

- **Richieste erogate e loro valorizzazione**: il report mostra solo le richieste erogate per unità operativa in quanto, per via della non additività delle misure, se si elimina dal report il raggruppamento delle unità operative non si ottengono le richieste per il dipartimento.

- **Prestazioni e referti in ritardo**: una prestazione o un referto è in ritardo quando viene erogato dopo la data prevista.

Il DataSource a cui puntano i dati è *PagodaVertica*. Le misure mostrate si riferiscono ad un determinato anno che viene selezionato dall'utente all'apertura del documento da una lista di tutti gli anni di refertazione disponibili. Al parametro di ingresso non è associato alcun valore di default e l'anno selezionato dall'utente verrà visualizzato in cima al documento.

La parte centrale del documento è una tabella pivot dove vengono mostrate, all'apertura di questo, le misure di interesse per unità operativa. Le dimensioni su cui è possibile modificare i filtri e i raggruppamenti del report sono: le UOP, i laboratori, l'anno di refertazione, l'origine, il tipo di reparto (che presenta solo i valori interno ed esterno), il reparto e i medici. Le procedure per filtrare e raggruppare sono le stesse descritte per il documento "Confronto richieste per laboratorio".

Il menu che serve per modificare la tabella pivot è uguale a quello del capitolo precedente e le varie funzionalità in esso riportate assumono lo stesso significato.

### 4.1.7 Documento 7: "Confronto richieste per provenienza"

L'ultimo documento che viene preso in considerazione nella sezione "Statistiche richieste" del progetto Pagoda è ancora un documento OLAP.

Le misure mostrate nel documento sono:

- **Richieste erogate e loro valorizzazione**: il report mostra le sole richieste erogate per dipartimento.

- **Prestazioni e referti in ritardo**: una prestazione o un referto vengono erogati in ritardo se vengono prodotti dopo la data prevista.

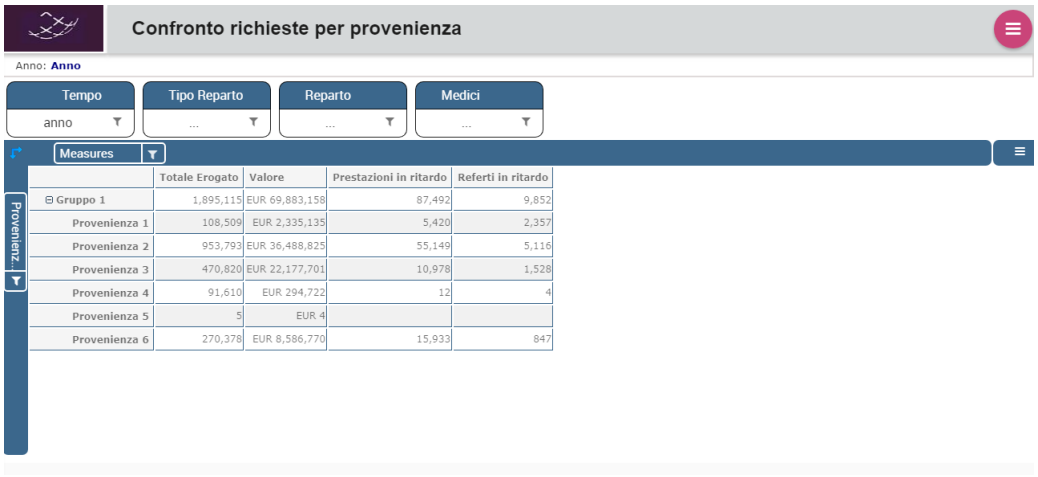


Figura 4.26: OLAP "Confronto richieste per provenienza"

Il DataSource di riferimento è *PagodaVertica*. Il parametro di ingresso è ancora una volta l'anno di refertazione che viene scelto dall'utente all'apertura del report da una lista degli anni disponibili ed è leggibile nel riquadro sopra la tabella pivot.

La tabella pivot mostra le misure per dipartimento e per origine. I filtri e i raggruppamenti possono essere fatti rispetto alle seguenti dimensioni: provenienza, anno di refertazione, tipo di reparto, reparto e medico. Il menu e le funzionalità che permettono di modificare la tabella sono le stesse degli altri documenti OLAP.

## 4.2 Porting di un documento data mining

La parte di lavoro che verrà qui presentata si pone come collegamento tra il lavoro descritto nelle pagine precedenti e quello che verrà descritto nell'ultimo paragrafo di questo capitolo. Si tratta sempre di un porting da SpagoBI a Knowage ma questa volta non di cruscotti o documenti OLAP ma di un documento data mining. Knowage contiene infatti anche un motore data mining che permette di integrare nel BI software script R o Python tramite l'utilizzo

di uno script XML. Questo motore fa sì che anche utenti non esperti siano in grado di visualizzare i risultati prodotti tramite il codice scritto; l'output del documento, infatti, verrà sempre ritornato dallo script R o Python.

Il documento riguarda la sezione "Estrazione risultati" del progetto Pagoda che si occupa di presentare i risultati delle varie analisi in base alle richieste di estrazione degli utenti autorizzati.


Figura 4.27: Cockpit "Elenco ultime 10 estrazioni"

Se si accede a questa sezione il primo documento che si apre è un cruscotto: questo illustra le ultime dieci estrazioni richieste dall'utente che ha effettuato il login nella dashboard, ovviamente se sono presenti. In particolare è una tabella che contiene, su ogni riga, il nome del richiedente l'estrazione, la data in cui è stata fatta, il codice e la descrizione dell'analisi di quella richiesta, un'eventuale nota e lo stato, che indica se l'estrazione è stata eseguita, archiviata, inserita o se c'è stato un errore. La colonna record indica la quantità di risultati che sono presenti per quell'estrazione.

Le ultime tre colonne, invece, sono diverse dalle precedenti in quanto a prima vista non mostrano alcuna informazione sulla richiesta ma costituiscono delle cross navigation su altri documenti. È possibile infatti cliccare su una di queste celle, in particolare solo su quelle che contengono l'icona, e accedere ai documenti di interesse.

La colonna "Parametri" ritorna il seguente cruscotto che illustra il dettaglio della richiesta di estrazione della riga selezionata.


Figura 4.28: Cockpit "Dettaglio estrazione richiesta"

La Figura 4.28 illustra una tabella riassuntiva dei parametri e delle caratteristiche della richiesta selezionata.

La colonna "Risultati" permette di accedere a un altro cruscotto. Quest'ultimo contiene una tabella che riporta i dettagli di tutti i risultati legati alla riga e quindi alla richiesta selezionata. In particolare ogni riga è un risultato e, per ogni risultato, si hanno i dettagli del reparto (tipo di reparto, codice, descrizione e laboratorio di accettazione), i valori della nota clinica e del quesito diagnostico (se presenti), i dettagli del paziente (come sesso e età) e, infine, il risultato dell'analisi effettuata con i relativi dettagli del laboratorio e dell'analisi. Naturalmente i parametri finora descritti vengono mostrati solo se presenti per quella richiesta. Il numero di righe della tabella è pari al valore presente nella colonna "Record". La cross navigation è inoltre abilitata solo se il valore di "stato estrazione" è uguale a "eseguito", e quindi l'estrazione è stata eseguita ma non ancora archiviata, e se il numero di record per quell'estrazione è inferiore a 300.000, altrimenti i tempi di caricamento del documento sarebbero troppo lunghi.

Infine, la colonna "Grafico" permette di accedere al documento data mining. In questo caso, nel file XML, è stato inserito uno script R che, grazie a una funzione, permette di creare un grafico a scelta dell'utente e con i pa-

ramatri del grafico anch'essi scelti dall'utente. All'apertura del documento colui che ha effettuato il login deve selezionare il tipo di grafico tra: boxplot, radar e scatterplot. Una volta selezionato questo si deve passare alla scelta dei parametri per l'asse X e per l'asse Y, anch'essi a discrezione dell'utente. Le opzioni "Tipo Grafico", "Asse X" e "Asse Y" descritte finora sono obbligatorie e sono sufficienti per la visualizzazione del grafico. È però possibile personalizzarlo ulteriormente selezionando dei valori per gli altri paramatri a disposizione. Ad esempio si può scegliere il nome degli assi, i valori di patologicità e normalità minima e massima, l'operazione da effettuare sui valori dell'asse Y per il grafico radar. Come nel caso della colonna "Risultati", anche per questa colonna non sempre è possibile accedere al documento data mining. La cross navigation infatti è abilitata solo sulle righe delle estrazioni con stato uguale a "eseguito".

L'ultima cross navigation definita sul documento di Figura 4.27 è quella che riguarda l'archivio delle estrazioni. Cliccando infatti sull'immagine a fianco di "Archivio estrazioni" si apre un cruscotto analogo a quello di Figura 4.27.

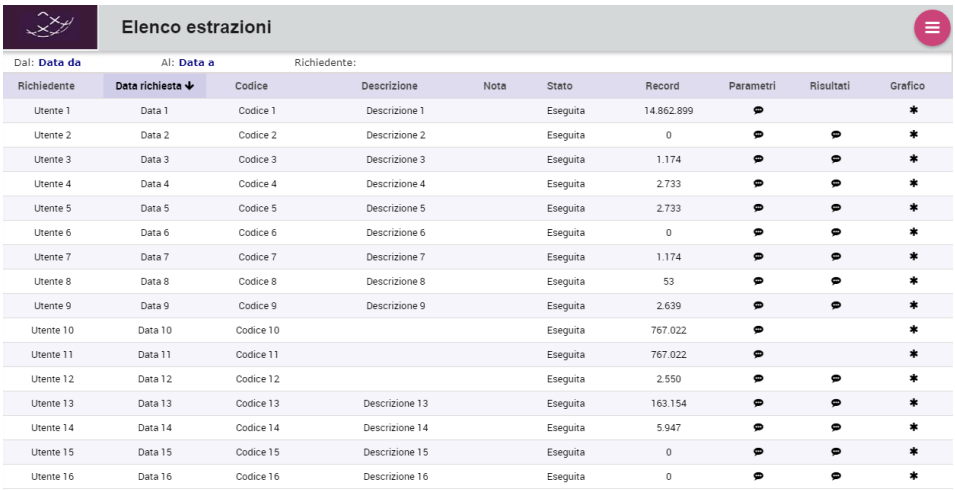


Figura 4.29: Cockpit "Elenco estrazioni"

Il documento mostra l'elenco di tutte le estrazioni che sono state fatte all'in-

terno delle date selezionate; all'apertura di questo occorre quindi selezionare la data di inizio e di fine delle estrazioni che si desidera visualizzare. È possibile anche specificare gli utenti che hanno fatto le estrazioni in modo da visualizzare quelle fatte all'interno delle date scelte e da quei soli utenti selezionati. Se non si seleziona nulla per quest'ultimo parametro verranno visualizzate le estrazioni fatte da tutti gli utenti abilitati a farle. Nel riquadro in alto del cockpit è possibile leggere i filtri abilitati. Per quanto riguarda la struttura della tabella, e quindi le colonne e le varie cross navigation, funziona nello stesso modo di quella descritta per il cruscotto di Figura 4.27.

## 4.3 Analisi con R

Prima di analizzare il lavoro fatto occorre capire quali siano le esigenze del cliente. Sono stati formulati due problemi da prendere in considerazione:

1. verificare se le soglie di riferimento sono corrette in relazione all'evoluzione e al cambiamento della popolazione nel corso del tempo;

2. verificare se l'essere esente o meno da ticket sia correlato alla patologicità del risultato.

Prima di affrontare nel dettaglio questi problemi è stato necessario estrarre i dati di interesse, in quanto ancora non esistevano raggruppati in un'unica tabella. Tramite un'apposita query sono stati unite le informazioni provenienti da tabelle diverse e ne è stata creata una apposita all'interno del DataSource *Vertica*. Di seguito viene riportata la query formulata per l'estrazione.

```
SELECT t1.DATA_ACCETTAZIONE, t4.CODICE_RICHIESTA, t4.CD_ANALISI,
t4.DESCR_ANALISI, t4.CD_RISULTATO, t4.DESCR_risultato,
t4.CODICE_CAMPIONE, t4.NUM_PRE_CAMPIONE, t1.PZ_COGNOME_NOME,
case when t1.PZ_SESSO='M' then 0 else 1 end PZ_SESSO,
AGE_IN_YEARS(t1.DATA_ACCETTAZIONE,t1.PZ_DT_NASCITA) PZ_ETA_ACC,
t1.PZ_DT_NASCITA, cm2.DESCRIZIONE_COMUNE PZ_CMN_RESIDENZA,
```

```
t1.CD_CONVENZIONE, cm3.DESCRIZIONE_COMUNE PZ_CMN_DOMICILIO,
t1.DESCR_CONVENZIONE, cm1.DESCRIZIONE_COMUNE PZ_CMN_NASCITA,
t1.QUESITO_DIAGNOSTICO, case when t1.QUESITO_DIAGNOSTICO is not
null then 1 else 0 end FLG_QUESITO_DIAGNOSTICO, t1.NOTA_CLINICA,
case when t1.NOTA_CLINICA is not null then 1 else 0 end
FLG_NOTA_CLINICA, t4.VALORE_NUMERICO, t4.FLG_NORMALE_PATOL,
t4.DATA_PRIMA_REFERTAZIONE_RISULTATO, ir.MINIMO_NORMALE,
ir.MASSIMO_NORMALE, ir.MINIMO_PATOLOGICO, ir.MASSIMO_PATOLOGICO,
ir.MINIMO_ACCETTABILE, ir.MASSIMO_ACCETTABILE, ir.INTERVALLO,
ir.CODICE_METODICA
FROM pagoda.T1_RICHIESTE t1, pagoda.T4_RISULTATI t4,
pagoda.PD_INTERVALLO_RISULTATI ir, pagoda.PA_COMUNI cm1,
pagoda.PA_COMUNI cm2, pagoda.PA_COMUNI cm3
WHERE t1.CODICE_RICHIESTA=t4.CODICE_RICHIESTA and
t1.PZ_CMN_NASCITA=cm1.CODICE_COMUNE and t1.PZ_CMN_RESIDENZA=
cm2.CODICE_COMUNE and t1.PZ_CMN_DOMICILIO=cm3.CODICE_COMUNE
and t4.CODICE_RICHIESTA=ir.CODICE_RICHIESTA and
t4.CD_ANALISI=ir.CODICE_ANALISI and t4.NUM_PRE_CAMPIONE=
ir.NUMERO_PRELIEVO_CAMPIONE and t4.CODICE_CAMPIONE=ir.CD_CAMPIONE
and t4.CD_RISULTATO=ir.CODICE_RISULTATO and t4.VALORE_NUMERICO
is not null and t1.TIPO_PAZIENTE='1' and t1.FLG_PAZIENTE_PROVA='0'
and t1.DATA_ACCETTAZIONE >= TO_DATE('01/01/2013', 'dd/mm/yyyy')
and t1.DATA_ACCETTAZIONE < TO_DATE('31/12/2017', 'dd/mm/yyyy')+1
```

Vengono estratti i dati dei pazienti (sesso, età, comune di provenienza e residenza), quelli relativi all'analisi (valore numerico del risultato, codice dell'analisi, flag di patologicità, soglie per identificare se un valore è patologico o meno) e quelli per l'esenzione (il codice e la descrizione dell'esenzione). Sono presenti tre diverse tipologie di soglie: il minimo e il massimo normali vengono utilizzati per identificare se un valore è patologico o meno, il minimo e massimo patologico indicano rispettivamente i valori patologici minimo e massimo registrati, il minimo e massimo accettabile rappresentano le soglie

entro le quali un valore è considerato accettabile (se il valore si trova al di fuori di queste probabilmente c'è stato un errore). I dati vengono poi filtrati per cinque anni, al fine di non estrarre un campione troppo numeroso e quindi difficile da elaborare; i pazienti considerati sono solo quelli esterni (t1.TIPO_PAZIENTE='1') e vengono quindi escluse le analisi di pazienti già ricoverati in quanto andrebbero a influenzare troppo il risultato finale. La tabella così creata è costituita da circa 10 mln di righe.

Per quanto riguarda il codice R scritto, dopo aver caricato le librerie necessarie ed aver effettuato la connessione al data source e al driver di Vertica, è stato necessario filtrare ulteriormente i dati in quanto, a causa della macchina a disposizione e del fatto che R tiene tutto in memoria, non sarebbe stato possibile elaborare 10 mln di righe. Tramite quindi un'interrogazione sulla nuova tabella generata, è stato creato un data set su cui effettuare le prime analisi. Il filtro che si è scelto di adottare è relativo alle analisi di laboratorio e sono state selezionate solo tre tipi di analisi: Bilirubina totale e frazionata (che include due risultati: bilirubina totale e bilirubina frazionata, sostanze prodotte dall'organismo durante la degradazione dell'emoglobina), TSH (ormone tireostimolante secreto dall'adenoipofisi) e Potassio. La scelta è stata dettata principalmente dalla quantità di dati presenti in ognuna di queste analisi in modo tale che non fossero né troppo pochi né troppi. In questo modo il data set è stato ridotto a poco più di 2 mln di righe e ogni riga corrisponde a una particolare analisi per un giorno e un paziente. È stato inoltre necessario eliminare le righe non corrette, ovvero quelle il cui risultato numerico era al di fuori delle soglie sia di normalità che di patologicità e nonostante questo veniva segnalato come valore non patologico. Tutti i risultati che verranno presentati di seguito possono essere ottenuti allo stesso modo aggiungendo o modificando i tipi di analisi di laboratorio considerate.

Per quanto riguarda il primo problema proposto, si è focalizzata l'attenzione sullo studio dell'evoluzione della distribuzione dei risultati delle analisi di laboratorio nel corso del tempo. Le diverse analisi devono essere considerate separatamente in quanto i valori medi associati a ciascuna di esse possono

essere molto diversi e le analisi sono indipendenti tra di loro. Prima di passare ad analizzare le distribuzioni nel corso degli anni (come e se sono variate), sono stati disegnati i seguenti boxplot per capire l'andamento generale dei valori delle analisi.

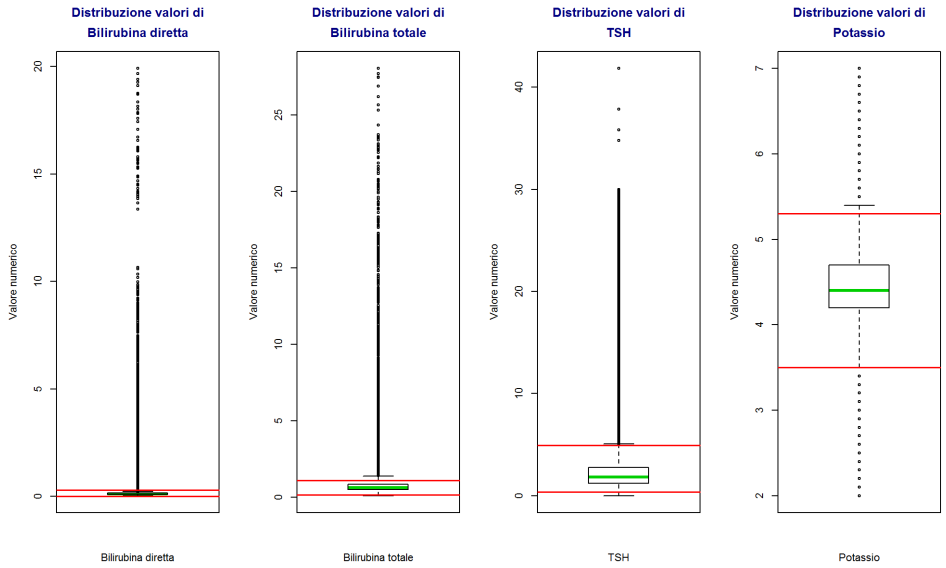


Figura 4.30: Boxplot valori numerici per risultato

Insieme ai boxplot sono state plottate in rosso le soglie di normalità. Come si può vedere dal grafico gli outliers corrispondono ai valori patologici dell'analisi, cosa comprensibile in quanto i valori che risultano essere patologici sono sempre meno dell'8% rispetto alla totalità dei valori registrati.

Per ogni analisi vengono poi disegnati dei boxplot, uno per ogni anno, per analizzare se e come è cambiata la popolazione dei risultati. Anche in questi grafici le linee rosse corrispondono alle soglie di normalità che rimangono fisse negli anni. I plot riportati di seguito sono uno zoom dei boxplot: è infatti interessante, ai fini dell'indagine, vedere soprattutto la zona circostante le soglie.

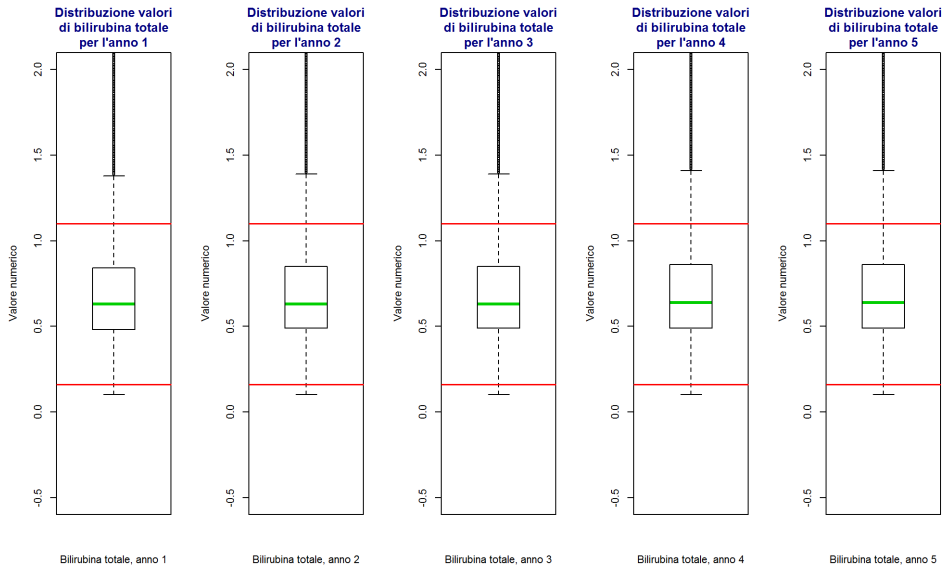


Figura 4.31: Boxplot valori di Bilirubina totale per ogni anno

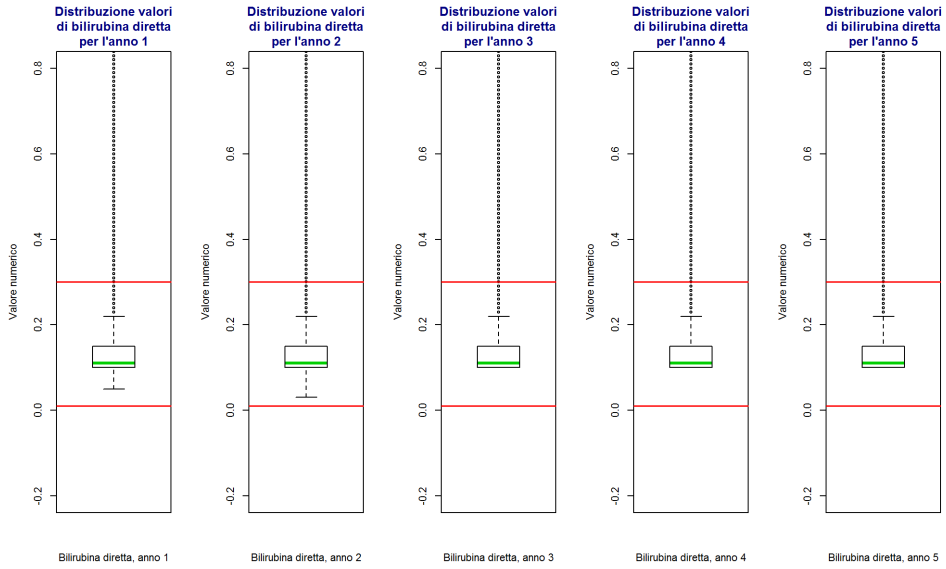


Figura 4.32: Boxplot valori di Bilirubina diretta per ogni anno

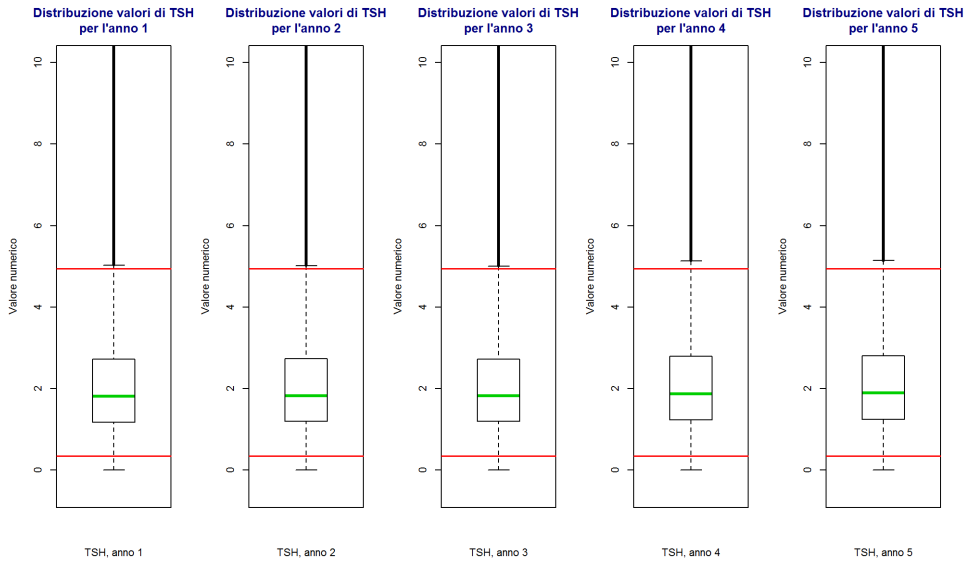


Figura 4.33: Boxplot valori di TSH per ogni anno

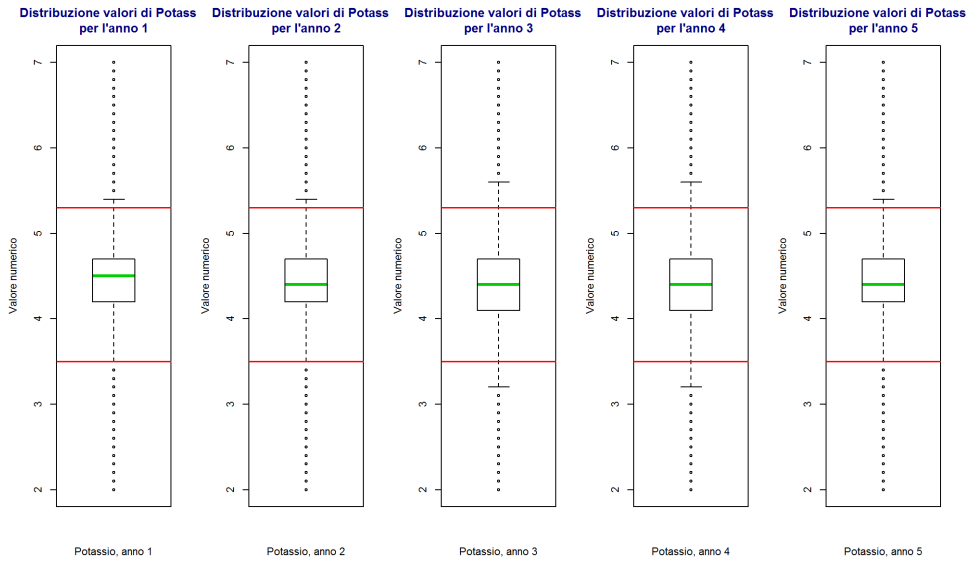


Figura 4.34: Boxplot valori di Potassio per ogni anno

Guardando questi plot è possibile dare una risposta iniziale al primo problema sottoposto dal cliente. Si può infatti vedere che, per tutte e tre le analisi

mediche, non vi è un cambiamento significativo nella distribuzione dei risultati e la mediana e l'area dei box sono molto simili negli anni. Si arriva quindi a concludere che è corretto mantenere le stesse soglie nel corso del tempo nonostante il target popolazione sia evoluto e cambiato. Tuttavia questa è solo una conclusione preliminare al problema posto. Per avere la certezza che sia corretta occorrerebbe avere a disposizione un orizzonte temporale molto più ampio rispetto a quello che si ha a disposizione: sarebbe interessante effettuare la stessa analisi su un orizzonte, ad esempio, di 50 anni. Occorrerebbe inoltre vedere se c'è una correlazione tra le variazioni dei risultati nel tempo e l'evoluzione della popolazione, tuttavia i dati a disposizione sui pazienti sono troppo riduttivi per effettuare questa analisi: sarebbe stato utile, ad esempio, avere l'indicazione se il paziente sia immigrato o meno anziché avere a disposizione unicamente i dati sull'età e il sesso del paziente.

Rivolgiamo ora l'attenzione al secondo problema formulato dal cliente e riguardante le esenzioni. Come analisi preliminare è stato fatto un test di Fisher per verificare se le popolazioni dei valori dei pazienti esenti e di quelli non esenti fossero omogenee. Per fare questo è stato necessario creare un secondo data set costituito da una tabella due per due: nella prima riga vengono presi in considerazione solo i pazienti con esenzione e sulle due colonne sono calcolate rispettivamente le quantità totali di valori normali e patologici, la seconda riga contiene le stesse informazioni ma riguardo ai pazienti non esenti. Per riconoscere se un paziente è dotato di una qualche esenzione è stata utilizzata la colonna CD_CONVENZIONE della tabella di partenza: se un paziente non è dotato di alcuna esenzione il valore della colonna è pari a SSN. Tuttavia nell'analisi sono stati considerati non esenti anche i pazienti dotati di esenzioni, ad esempio, per il reddito o comunque non legate a nessuna malattia o patologia particolare. Come nel primo data set, anche in questo caso sono state eliminate le righe non corrette. Applicando quindi il test di Fisher tramite la funzione R `fisher.test` al data set appena creato, si ottiene il seguente risultato: `p-value < 2.2e-16`. Questo ci porta dunque a rifiutare l'ipotesi nulla del test, che sostiene l'omogeneità delle due

popolazioni. Si può allora concludere che tra le due popolazioni, dei valori dei pazienti esenti e non esenti, ci sia una differenza significativa e che quindi l'esenzione è correlata alla patologicità. Tuttavia è più corretto usare il test di Fisher quando si ha un campione poco numeroso. Nel caso in cui si abbiano a disposizione molti dati, come in questa situazione, è meglio utilizzare il test Chi quadrato. Applicando allora la funzione `chisq.test` ai dati si ottiene il seguente risultato: `p-value < 2.2e-16`. Si ricava una conferma delle conclusioni a cui si era giunti tramite il test di Fisher; infatti, poiché il p-value è molto basso, si è portati a rifiutare l'ipotesi nulla di omogeneità.

Dopo essere arrivati a questa prima conclusione ci si è concentrati ad analizzare meglio in che modo l'essere esente o meno incida sull'avere un valore al di fuori delle soglie di normalità. Per fare questo sono stati utilizzati dati aggregati: si ha una riga per gli esenti e una per i pazienti non dotati di alcuna esenzione, la prima colonna contiene il numero dei pazienti appartenenti a quella categoria (esenti o non esenti) mentre nella seconda si ha la quantità di valori patologici. Anche in questo caso, un paziente viene considerato esente solo se lo è per una particolare malattia o patologia. Viene poi applicato il modello generale lineare con famiglia binomiale, o anche chiamato regressione logistica, nel seguente modo:

```
successo = NUM_PATOLOGICI
fallimento = NUM_PAZIENTI - NUM_PATOLOGICI
model1 <- glm(cbind(successo,fallimento) ~ ESENTI, family = binomial)
```

Attraverso un modello di regressione logistica si vuole investigare l'effetto dell'essere esente o meno sulla probabilità che il valore dell'analisi medica possa essere patologico. La variabile risposta in questo caso è la seguente: (numero di successi, numero di fallimenti), dove con successo si intende quando il valore risulta patologico. Quest'ultima dipende unicamente dal fattore ESENTI che può essere true o false. Il parametro di interesse è dunque la probabilità di essere patologico. Se si applica la funzione `summary` al modello appena creato, R fornisce i risultati in termini del parametro naturale $\theta$.

Tuttavia, in questo modo, i risultati sono difficili da interpretare ed è quindi preferibile guardare ai coefficienti esponenziati:

| | Estimate | 2.5 % | 97.5 % |
|---|---|---|---|
| (Intercept) | 0.06290285 | 0.06240206 | 0.06340646 |
| ESENTItrue | 1.40066911 | 1.38625574 | 1.41524121 |

Il coefficiente dell'esenzione è l'odds-ratio di esenti versus non esenti e viene interpretato nel seguente modo: le possibilità che il valore sia patologico se si è esenti sono 1.4 volte le possibilità che sia patologico se non si è esenti. Per quanto riguarda invece i parametri di interesse si hanno i seguenti risultati:

- la probabilità stimata di essere patologico se non esente è 0.06 (SE: 0.00023);
- la probabilità stimata di essere patologico se esente è 0.08 (SE: 0.00025).

Per valutare la bontà del modello viene confrontato il modello saturato, che in questo caso è equivalente a quello proposto poiché il numero di osservazioni è uguale al numero dei parametri, con il modello nullo. Calcolando il p_value si ottiene un risultato pari a 0 che porta a concludere che il modello proposto è migliore di quello saturato.

Quanto detto finora sulla regressione logistica conferma le conclusioni a cui si era giunti tramite il test di Fisher, ovvero che l'esenzione incide sulla patologicità. Tuttavia l'odds-ratio è di poco superiore a 1, di conseguenza si può anche affermare che il fatto di essere esenti non dà la certezza di essere patologici. Quanto trovato risolve solo in parte il problema esposto dal cliente. Sarebbe interessante approfondire quale tipologia di esenzione è più soggetta ad essere associata a valori patologici. Per fare questo si potrebbe, ad esempio, effettuare un'analisi tramite cluster.

Arrivati alla conclusione che l'esenzione incide sulla patologicità, si è voluto vedere se ci fossero anche altri parametri che incidono su questa, quali l'età e il sesso del paziente. Inizialmente sono state fatte alcune analisi grafiche preliminari per osservare come fossero distribuite le due popolazioni di

pazienti esenti e non esenti. Tramite una query è stato estratto l'ultimo data set di interesse. In questo caso i dati non sono aggregati ma ogni riga corrisponde ad un'analisi di un paziente e in un particolare giorno. È stata aggiunta una colonna contenente l'informazione sulla presenza di esenzione per il paziente: la colonna ESENTI contiene quindi il valore booleano si/no. I primi grafici che verranno presentati sono degli istogrammi: la variabile presa in considerazione è il valore numerico del risultato dell'analisi medica. Per ogni tipo di analisi di laboratorio sono stati disegnati due istogrammi sovrapposti: in blu sono plottati i valori relativi alla popolazione dei pazienti dotati di almeno un'esenzione, in verde invece sono plottati i valori per la popolazione di pazienti non esenti.

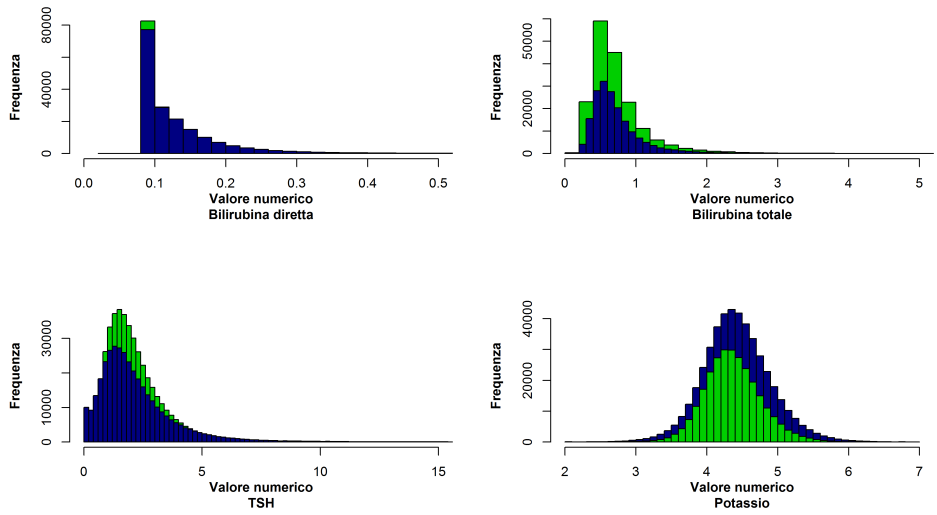


Figura 4.35: Istogrammi dei pazienti esenti e non divisi per tipo di analisi

Come si può vedere dai grafici sovrastanti, per tutti i tipi di analisi le distribuzioni delle due popolazioni seguono un andamento molto simile. In particolare si può vedere che i valori di bilirubina diretta seguono una distribuzione esponenziale, quelli di bilirubina totale e di TSH seguono una distribuzione gamma e quelli del potassio una distribuzione gaussiana.

Per mettere meglio a confronto le due distribuzioni sono stati disegnati anche dei boxplot, uno per la popolazione di pazienti esenti e l'altro per quella dei pazienti senza esenzione.

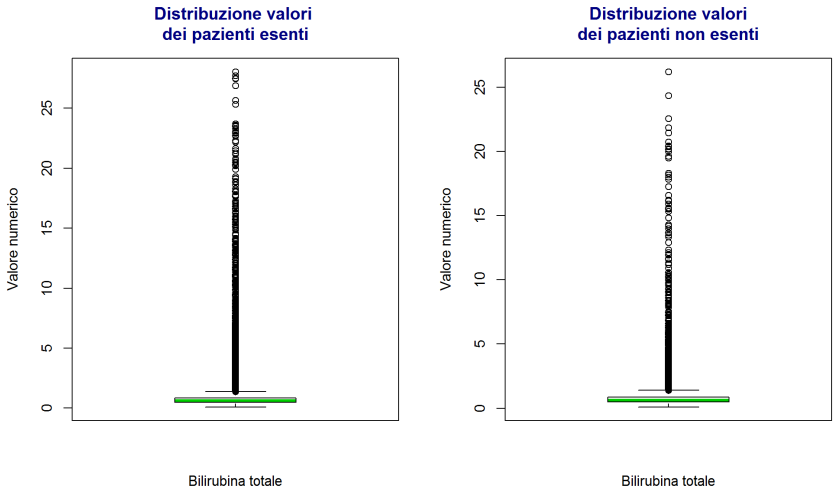


Figura 4.36: Boxplot valori di Bilirubina totale per i pazienti esenti e non

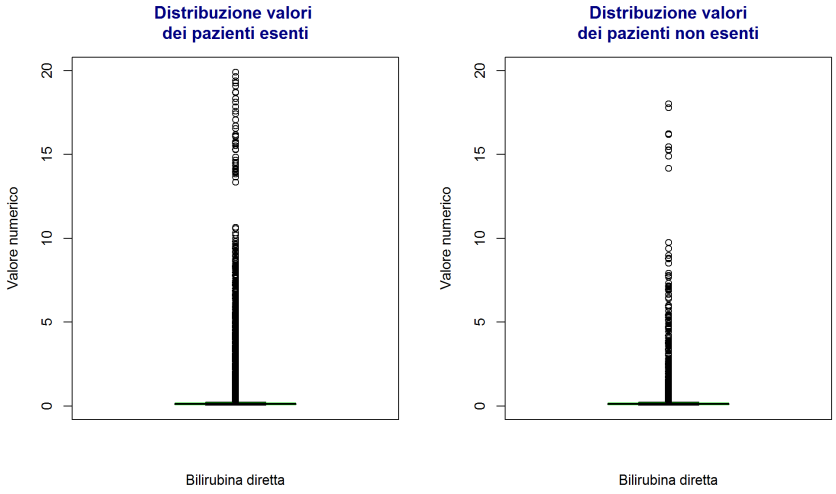


Figura 4.37: Boxplot valori di Bilirubina diretta per i pazienti esenti e non

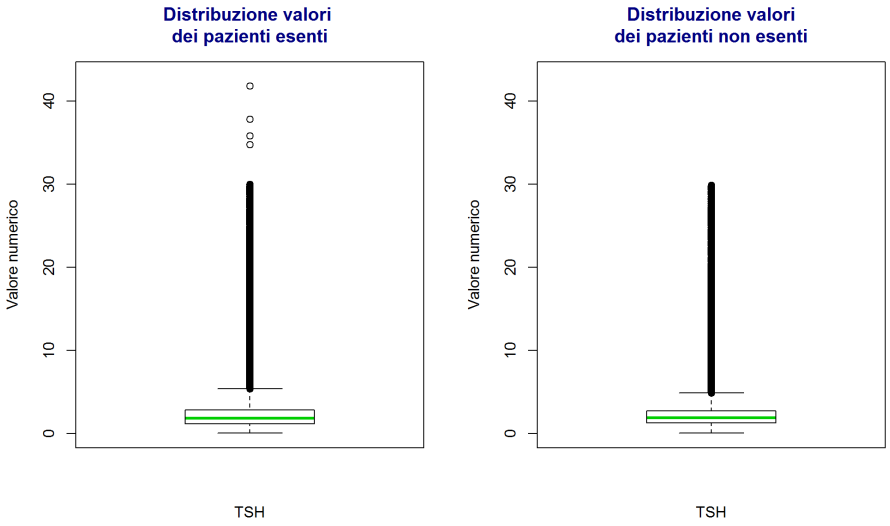


Figura 4.38: Boxplot valori di TSH per ogni i pazienti esenti e non

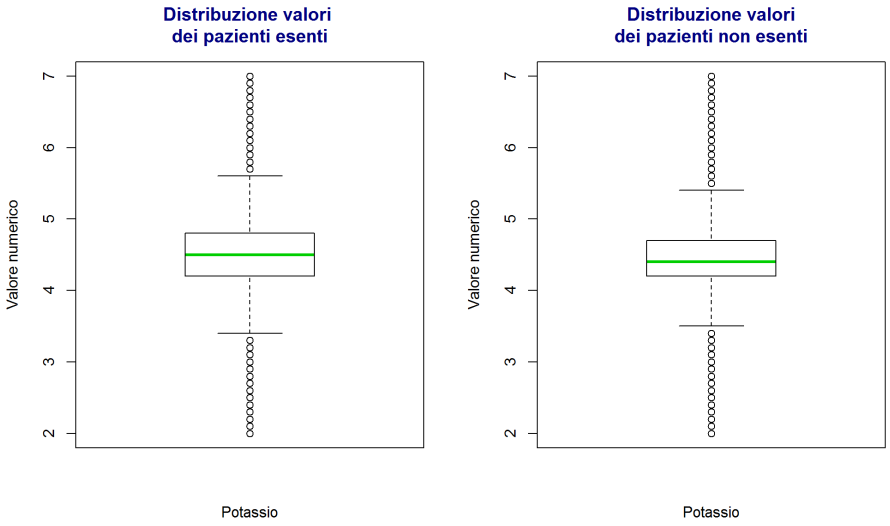


Figura 4.39: Boxplot valori di Potassio per ogni i pazienti esenti e non

Vi sono due boxplot per ogni tipo di analisi di laboratorio. Questi confermano quello evidenziato dagli istogrammi, ovvero che le due popolazioni hanno un andamento simile.

In seguito a queste analisi grafiche è stata nuovamente utilizzata la regressione logistica per creare un nuovo modello. Questa volta, anziché includere solamente il fattore di esenzione, sono stati inclusi come predittori anche il sesso, l'età e i comuni di nascita, residenza e domicilio del paziente. La variabile sesso assume il valore 1 se il paziente è una femmina e il valore 0 se il paziente è un maschio. Viene fittata una GLM per dati binari con il link logistico (binomial):

```
model2 <- glm(FLG_NORMALE_PATOL ~ PZ_ETA_ACC + PZ_SESSO +
PZ_CMN_DOMICILIO + PZ_CMN_NASCITA + PZ_CMN_RESIDENZA +
ESENTI, family=binomial)
```

La variabile risposta è il flag di patologicità che assume i valori 0 se il risultato rientra nelle soglie di normalità e 1 se non rientra in queste soglie. Anche in questo caso per interpretare i risultati si guarda ai coefficienti esponenziati:

| | Estimate | 2.5 % | 97.5 % | |
|---|---|---|---|---|
| (Intercept) | 0.0575220 | 0.05629714 | 0.0587794 | $***$ |
| PZ_ETA_ACC | 1.0025318 | 1.00222220 | 1.0028418 | $***$ |
| PZ_SESSO1 | 0.9128901 | 0.90349264 | 0.9223902 | $***$ |
| PZ_CMN_DOMICILIO | 0.9999995 | 0.99999864 | 1.0000005 | |
| PZ_CMN_NASCITA | 1.0000002 | 1.00000001 | 1.0000004 | $*$ |
| PZ_CMN_RESIDENZA | 1.0000007 | 0.99999973 | 1.0000015 | |
| ESENTIsi | 1.3362729 | 1.31979421 | 1.3529611 | $***$ |

Per quanto riguarda i comuni, è stato utilizzato il codice di avviamento postale in modo che potessero essere trattati come dei numeri interi. Se veniva utilizzato il nome, infatti, sarebbe risultato un fattore con troppi livelli e R non sarebbe riuscito a elaborarlo.

Da quanto riportato sopra è possibile arrivare alle seguenti conclusioni:

- quando l'età del paziente aumenta di un anno il rischio che il valore dell'analisi sia patologico è moltiplicato di 1.0025, questo risultato è statisticamente molto significativo, il valore del p-value del coefficiente stimato è infatti molto basso;
- quando il paziente è una donna il valore è moltiplicato di 0.913 (o equivalentemente diviso di 1.095); anche questo risultato è statisticamente molto significativo;
- il comune di domicilio e quello di residenza non sono statisticamente significativi, dato che il loro p-value è molto alto; il comune di nascita è invece più significativo rispetto agli altri ma comunque non quanto gli altri predittori;
- quando il paziente è dotato di una qualche esenzione il rischio che il valore sia patologico è moltiplicato di 1.3363, questo è un risultato statisticamente molto significativo (l'esenzione incide sulla patologicità) e conferma le conclusioni a cui si era giunti in precedenza tramite il primo modello.

Passando poi allo studio della qualità del modello, si utilizza il test chi quadrato per confrontare il modello nullo con il modello saturato prima e quello saturato con quello proposto dopo. Il modello proposto è quello esposto finora, il modello nullo è quello che utilizza esclusivamente l'intercetta e, infine, quello proposto utilizza un numero di predittori pari alla numerosità del data set. Nel confronto tra il modello nullo e il modello saturato si ottiene un p-value pari a 1: il modello nullo è migliore rispetto a quello saturato. Questo risultato è comprensibile: sarebbe infatti impossibile creare un modello con più di 2 mln di predittori. Quando si confrontano il modello saturato con quello proposto si ottiene ancora un p-value pari a 1: in questo caso significa che il modello proposto è migliore rispetto a quello saturato. Di conseguenza il modello creato fitta bene i dati a disposizione.

### 4.3.1 Implementazione

Riporto di seguito il codice R utilizzato per le analisi sopra riportate.

```
wd <- "xxx"
setwd(wd)

library(DBI)
library(magrittr)
library(dplyr)
library(xtable)

drv <- xxx("xxx", classPath="xxx", " ")
con <- dbConnect(drv, "xxx", "xxx", "xxx")

dati <- dbGetQuery(con, "select YEAR(DATA_ACCETTAZIONE) as ANNO, PZ_SESSO,
PZ_ETA_ACC, CD_CONVENZIONE, DESCR_risultato, FLG_NORMALE_PATOL,
VALORE_NUMERICO, MINIMO_NORMALE, MASSIMO_NORMALE, MINIMO_PATOLOGICO,
MASSIMO_PATOLOGICO
from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI in ('109','600','162') and CODICE_RICHIESTA not in
(select CODICE_RICHIESTA from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI in ('109','600','162') and FLG_NORMALE_PATOL=0 and
(VALORE_NUMERICO<MINIMO_PATOLOGICO or VALORE_NUMERICO>MASSIMO_PATOLOGICO))")

head(dati)
attach(dati)
length(dati)
names(dati)
nrow(dati)
summary(dati)
str(dati)
```

```
png(file="Boxplot per cd_risultato.png", res = 200, width=10,height=6,
units = "in")
par(mfrow=c(1,length(levels(DESCR_risultato))))
for (i in 1:length(levels(DESCR_risultato))) {
  boxplot(VALORE_NUMERICO[DESCR_risultato==levels(DESCR_risultato)[i]],
  main=cbind("Distribuzione valori di\n",levels(DESCR_risultato)[i]),
  xlab=levels(DESCR_risultato)[i], ylab="Valore numerico", col.main="navy",
  medcol="green3", font.lab=2)
  abline(h=mean(MINIMO_NORMALE[DESCR_risultato==levels(DESCR_risultato)[i]]),
  col="red", lwd=1.7)
  abline(h=mean(MASSIMO_NORMALE[DESCR_risultato==levels(DESCR_risultato)[i]]),
  col="red", lwd=1.7)
}
dev.off()

# Boxplot per ogni anno per i risultati di Bilirubina totale
png(file="Boxplot bili per anno.png", res = 200, width=10,height=6,
units = "in")
par(mfrow=c(1,5))
boxplot(VALORE_NUMERICO[ANNO==1 & DESCR_risultato=="Bilirubina totale"],
main="Distribuzione valori\ndi bilirubina totale\nper l'anno 1",
xlab="Bilirubina totale, anno 1", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2, ylim=c(-0.5,2))
abline(h=mean(MINIMO_NORMALE[ANNO==1 & DESCR_risultato==
"Bilirubina totale"]), col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==1 & DESCR_risultato==
"Bilirubina totale"]), col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==2 & DESCR_risultato=="Bilirubina totale"],
main="Distribuzione valori\ndi bilirubina totale\nper l'anno 2",
xlab="Bilirubina totale, anno 2", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2, ylim=c(-0.5,2))
```

```
abline(h=mean(MINIMO_NORMALE[ANNO==2 & DESCR_risultato==
"Bilirubina totale"]), col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==2 & DESCR_risultato==
"Bilirubina totale"]), col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==3 & DESCR_risultato=="Bilirubina totale"],
main="Distribuzione valori\ndi bilirubina totale\nper l'anno 3",
xlab="Bilirubina totale, anno 3", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2, ylim=c(-0.5,2))
abline(h=mean(MINIMO_NORMALE[ANNO==3 & DESCR_risultato==
"Bilirubina totale"]), col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==3 & DESCR_risultato==
"Bilirubina totale"]), col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==4 & DESCR_risultato=="Bilirubina totale"],
main="Distribuzione valori\ndi bilirubina totale\nper l'anno 4",
xlab="Bilirubina totale, anno 4", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2, ylim=c(-0.5,2))
abline(h=mean(MINIMO_NORMALE[ANNO==4 & DESCR_risultato==
"Bilirubina totale"]), col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==4 & DESCR_risultato==
"Bilirubina totale"]), col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==5 & DESCR_risultato=="Bilirubina totale"],
main="Distribuzione valori\ndi bilirubina totale\nper l'anno 5",
xlab="Bilirubina totale, anno 5", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2, ylim=c(-0.5,2))
abline(h=mean(MINIMO_NORMALE[ANNO==5 & DESCR_risultato==
"Bilirubina totale"]), col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==5 & DESCR_risultato==
"Bilirubina totale"]), col="red", lwd=1.7)
dev.off()

# Boxplot per ogni anno per i risultati di Bilirubina diretta
```

```
png(file="Boxplot bili2 per anno.png", res = 200, width=10,height=6,
units = "in")
par(mfrow=c(1,5))
boxplot(VALORE_NUMERICO[ANNO==1 & DESCR_risultato=="Bilirubina diretta"],
main="Distribuzione valori\ndi bilirubina diretta\nper l'anno 1",
xlab="Bilirubina diretta, anno 1", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2, ylim=c(-0.2,0.8))
abline(h=mean(MINIMO_NORMALE[ANNO==1 & DESCR_risultato==
"Bilirubina diretta"]), col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==1 & DESCR_risultato==
"Bilirubina diretta"]), col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==2 & DESCR_risultato=="Bilirubina diretta"],
main="Distribuzione valori\ndi bilirubina diretta\nper l'anno 2",
xlab="Bilirubina diretta, anno 2", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2, ylim=c(-0.2,0.8))
abline(h=mean(MINIMO_NORMALE[ANNO==2 & DESCR_risultato==
"Bilirubina diretta"]), col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==2 & DESCR_risultato==
"Bilirubina diretta"]), col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==3 & DESCR_risultato=="Bilirubina diretta"],
main="Distribuzione valori\ndi bilirubina diretta\nper l'anno 3",
xlab="Bilirubina diretta, anno 3", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2, ylim=c(-0.2,0.8))
abline(h=mean(MINIMO_NORMALE[ANNO==3 & DESCR_risultato==
"Bilirubina diretta"]), col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==3 & DESCR_risultato==
"Bilirubina diretta"]), col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==4 & DESCR_risultato=="Bilirubina diretta"],
main="Distribuzione valori\ndi bilirubina diretta\nper l'anno 4",
xlab="Bilirubina diretta, anno 4", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2, ylim=c(-0.2,0.8))
```

```
abline(h=mean(MINIMO_NORMALE[ANNO==4 & DESCR_risultato==
"Bilirubina diretta"]), col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==4 & DESCR_risultato==
"Bilirubina diretta"]), col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==5 & DESCR_risultato=="Bilirubina diretta"],
main="Distribuzione valori\ndi bilirubina diretta\nper l'anno 5",
xlab="Bilirubina diretta, anno 5", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2, ylim=c(-0.2,0.8))
abline(h=mean(MINIMO_NORMALE[ANNO==5 & DESCR_risultato==
"Bilirubina diretta"]), col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==5 & DESCR_risultato==
"Bilirubina diretta"]), col="red", lwd=1.7)
dev.off()

# Boxplot per ogni anno per i risultati di TSH
png(file="Boxplot TSH per anno.png", res = 200, width=10, height=6,
units = "in")
par(mfrow=c(1,5))
boxplot(VALORE_NUMERICO[ANNO==1 & DESCR_risultato=="TSH"],
main="Distribuzione valori di TSH\nper l'anno 1", xlab="TSH, anno 1",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2,
ylim=c(-0.5,10))
abline(h=mean(MINIMO_NORMALE[ANNO==1 & DESCR_risultato=="TSH"]),
col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==1 & DESCR_risultato=="TSH"]),
col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==2 & DESCR_risultato=="TSH"],
main="Distribuzione valori di TSH\nper l'anno 2", xlab="TSH, anno 2",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2,
ylim=c(-0.5,10))
abline(h=mean(MINIMO_NORMALE[ANNO==2 & DESCR_risultato=="TSH"]),
```

```
col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==2 & DESCR_risultato=="TSH"]),
col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==3 & DESCR_risultato=="TSH"],
main="Distribuzione valori di TSH\nper l'anno 3", xlab="TSH, anno 3",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2,
ylim=c(-0.5,10))
abline(h=mean(MINIMO_NORMALE[ANNO==3 & DESCR_risultato=="TSH"]),
col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==3 & DESCR_risultato=="TSH"]),
col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==4 & DESCR_risultato=="TSH"],
main="Distribuzione valori di TSH\nper l'anno 4", xlab="TSH, anno 4",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2,
ylim=c(-0.5,10))
abline(h=mean(MINIMO_NORMALE[ANNO==4 & DESCR_risultato=="TSH"]),
col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==4 & DESCR_risultato=="TSH"]),
col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==5 & DESCR_risultato=="TSH"],
main="Distribuzione valori di TSH\nper l'anno 5", xlab="TSH, anno 5",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2,
ylim=c(-0.5,10))
abline(h=mean(MINIMO_NORMALE[ANNO==5 & DESCR_risultato=="TSH"]),
col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==5 & DESCR_risultato=="TSH"]),
col="red", lwd=1.7)
dev.off()

# Boxplot per ogni anno per i risultati di Potassio
png(file="Boxplot Potassio per anno.png", res = 200, width=10,height=6,
```

```
units = "in")
par(mfrow=c(1,5))
boxplot(VALORE_NUMERICO[ANNO==1 & DESCR_risultato=="Potassio"],
main="Distribuzione valori di Potassio\nper l'anno 1",
xlab="Potassio, anno 1", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2)
abline(h=mean(MINIMO_NORMALE[ANNO==1 & DESCR_risultato=="Potassio"]),
col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==1 & DESCR_risultato=="Potassio"]),
col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==2 & DESCR_risultato=="Potassio"],
main="Distribuzione valori di Potassio\nper l'anno 2",
xlab="Potassio, anno 2", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2)
abline(h=mean(MINIMO_NORMALE[ANNO==2 & DESCR_risultato=="Potassio"]),
col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==2 & DESCR_risultato=="Potassio"]),
col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==3 & DESCR_risultato=="Potassio"],
main="Distribuzione valori di Potassio\nper l'anno 3",
xlab="Potassio, anno 3", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2)
abline(h=mean(MINIMO_NORMALE[ANNO==3 & DESCR_risultato=="Potassio"]),
col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==3 & DESCR_risultato=="Potassio"]),
col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==4 & DESCR_risultato=="Potassio"],
main="Distribuzione valori di Potassio\nper l'anno 4",
xlab="Potassio, anno 4", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2)
abline(h=mean(MINIMO_NORMALE[ANNO==4 & DESCR_risultato=="Potassio"]),
```

```
col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==4 & DESCR_risultato=="Potassio"]),
col="red", lwd=1.7)
boxplot(VALORE_NUMERICO[ANNO==5 & DESCR_risultato=="Potassio"],
main="Distribuzione valori di Potassio\nper l'anno 5",
xlab="Potassio, anno 5", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2)
abline(h=mean(MINIMO_NORMALE[ANNO==5 & DESCR_risultato=="Potassio"]),
col="red", lwd=1.7)
abline(h=mean(MASSIMO_NORMALE[ANNO==5 & DESCR_risultato=="Potassio"]),
col="red", lwd=1.7)
dev.off()

detach(dati)

datiF <- dbGetQuery(con, "select (COUNT(*)-SUM(FLG_NORMALE_PATOL)) as
NUM_NO_PATOLOGICI, SUM(FLG_NORMALE_PATOL) as NUM_PATOLOGICI
from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI  in ('109','600','162') and CD_CONVENZIONE not in
('SSN', 'F01', 'RE1', 'E03', 'T01', 'P99', 'E02', 'RE4', 'PSU', 'E04',
'RE3', 'RE2', 'E99', 'I01', 'MSPM', 'SSA')and CODICE_RICHIESTA not in
(select CODICE_RICHIESTA from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI in ('109','600','162') and FLG_NORMALE_PATOL=0 and
(VALORE_NUMERICO<MINIMO_PATOLOGICO or VALORE_NUMERICO>MASSIMO_PATOLOGICO))
union
select (COUNT(*)-SUM(FLG_NORMALE_PATOL)) as NUM_NO_PATOLOGICI,
SUM(FLG_NORMALE_PATOL) as NUM_PATOLOGICI
from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI in ('109','600','162') and CD_CONVENZIONE in
('SSN', 'F01', 'RE1', 'E03', 'T01', 'P99', 'E02', 'RE4', 'PSU', 'E04',
'RE3', 'RE2', 'E99', 'I01', 'MSPM', 'SSA') and CODICE_RICHIESTA not in
```

```
(select CODICE_RICHIESTA from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI in ('109','600','162') and FLG_NORMALE_PATOL=0 and
(VALORE_NUMERICO<MINIMO_PATOLOGICO or VALORE_NUMERICO>MASSIMO_PATOLOGICO))")

fisher.test(datiF)
chisq.test(datiF)

datiBin <- dbGetQuery(con, "select COUNT(*) as NUM_PAZIENTI,
SUM(FLG_NORMALE_PATOL) as NUM_PATOLOGICI, 'true' as ESENTI
from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI in ('109','600','162') and CD_CONVENZIONE not in
('SSN', 'F01', 'RE1', 'E03', 'T01', 'P99', 'E02', 'RE4', 'PSU', 'E04',
'RE3', 'RE2', 'E99', 'I01', 'MSPM', 'SSA') and CODICE_RICHIESTA not in
(select CODICE_RICHIESTA from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI in ('109','600','162') and FLG_NORMALE_PATOL=0 and
(VALORE_NUMERICO<MINIMO_PATOLOGICO or VALORE_NUMERICO>MASSIMO_PATOLOGICO))
union
select COUNT(*) as NUM_PAZIENTI, SUM(FLG_NORMALE_PATOL) as NUM_PATOLOGICI,
'false' as ESENTI
from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI  in ('109','600','162') and CD_CONVENZIONE in
('SSN', 'F01', 'RE1', 'E03', 'T01', 'P99', 'E02', 'RE4', 'PSU', 'E04',
'RE3', 'RE2', 'E99', 'I01', 'MSPM', 'SSA') and CODICE_RICHIESTA not in
(select CODICE_RICHIESTA from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI in ('109','600','162') and FLG_NORMALE_PATOL=0 and
(VALORE_NUMERICO<MINIMO_PATOLOGICO or VALORE_NUMERICO>MASSIMO_PATOLOGICO))")

attach(datiBin)
str(datiBin)
table(NUM_PAZIENTI,ESENTI)
table(NUM_PATOLOGICI,ESENTI)
```

```
successo = NUM_PATOLOGICI
fallimento = NUM_PAZIENTI - NUM_PATOLOGICI
model1 <- glm(cbind(successo,fallimento) ~ ESENTI, family = binomial)
summary(model1)

confint(model1)
exp(coef(model1))
exp(confint(model1))
predict(model1, type="response", se.fit=T)
1/exp(coef(model1)[2])
1/exp(confint(model1)[2,])
1-pchisq(model1$null.deviance, model1$df.null)

detach(datiBin)

datiBern <- dbGetQuery(con, "select YEAR(DATA_ACCETTAZIONE) as ANNO, PZ_SESSO,
PZ_ETA_ACC, PZ_CMN_DOMICILIO, PZ_CMN_NASCITA, PZ_CMN_RESIDENZA,
FLG_NORMALE_PATOL, VALORE_NUMERICO, CD_CONVENZIONE, DESCR_risultato,
case when CD_CONVENZIONE not in ('SSN', 'F01', 'RE1', 'E03', 'T01', 'P99',
'E02', 'RE4', 'PSU', 'E04', 'RE3', 'RE2', 'E99', 'I01', 'MSPM', 'SSA') then
'si' else 'no' end ESENTI
from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI in ('109','600','162') and CODICE_RICHIESTA not in
(select CODICE_RICHIESTA from pagoda.PD_RICHIESTE_INTERVALLI
where CD_ANALISI in ('109','600','162') and FLG_NORMALE_PATOL=0 and
(VALORE_NUMERICO<MINIMO_PATOLOGICO or VALORE_NUMERICO>MASSIMO_PATOLOGICO))")

attach(datiBern)
str(datiBern)
PZ_SESSO <- as.factor(PZ_SESSO)
```

```
FLG_NORMALE_PATOL <- as.factor(FLG_NORMALE_PATOL)

png(file="Istogramma pazienti esenti e non.png", res = 400, width=10,
height=6, units = "in")
par(mfrow=c(2,2))
hist(VALORE_NUMERICO[DESCR_risultato=="Bilirubina diretta" & ESENTI=="no"],
breaks=1000, main = "", xlab=cbind("Valore numerico\nBilirubina diretta"),
ylab = "Frequenza", col="green3", font.lab=2, xlim=c(0,0.5))
hist(VALORE_NUMERICO[DESCR_risultato=="Bilirubina diretta" & ESENTI=="si"],
breaks=1000, col="navy", add=T)
hist(VALORE_NUMERICO[DESCR_risultato=="Bilirubina totale" & ESENTI=="no"],
breaks=160, main = "", xlab=cbind("Valore numerico\nBilirubina totale"),
ylab = "Frequenza", col="green3", font.lab=2, xlim=c(0,5))
hist(VALORE_NUMERICO[DESCR_risultato=="Bilirubina totale" & ESENTI=="si"],
breaks=200, col="navy", add=T)
hist(VALORE_NUMERICO[DESCR_risultato=="TSH" & ESENTI=="no"], breaks=160,
main = "", xlab=cbind("Valore numerico\nTSH"), ylab = "Frequenza",
col="green3", font.lab=2, xlim=c(0,15))
hist(VALORE_NUMERICO[DESCR_risultato=="TSH" & ESENTI=="si"], breaks=200,
col="navy", add=T)
hist(VALORE_NUMERICO[DESCR_risultato=="Potassio" & ESENTI=="si"], breaks=50,
main = "", xlab=cbind("Valore numerico\nPotassio"), ylab = "Frequenza",
col="navy", font.lab=2)
hist(VALORE_NUMERICO[DESCR_risultato=="Potassio" & ESENTI=="no"], breaks=50,
col="green3", add=T)
dev.off()

png(file="Boxplot Bilirubina totale per esenzione.png", res = 200, width=10,
height=6, units = "in")
par(mfrow=c(1,2))
boxplot(VALORE_NUMERICO[DESCR_risultato=="Bilirubina totale" & ESENTI=='si'],
```

```
main="Distribuzione valori \ndei pazienti esenti", xlab="Bilirubina totale",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2)
boxplot(VALORE_NUMERICO[DESCR_risultato=="Bilirubina totale" & ESENTI=='no'],
main="Distribuzione valori\ndei pazienti non esenti",
xlab="Bilirubina totale", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2)
dev.off()
png(file="Boxplot Bilirubina diretta per esenzione.png", res = 200, width=10,
height=6, units = "in")
par(mfrow=c(1,2))
boxplot(VALORE_NUMERICO[DESCR_risultato=="Bilirubina diretta" & ESENTI=='si'],
main="Distribuzione valori \ndei pazienti esenti", xlab="Bilirubina diretta",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2,
ylim=c(0,20))
boxplot(VALORE_NUMERICO[DESCR_risultato=="Bilirubina diretta" & ESENTI=='no'],
main="Distribuzione valori\ndei pazienti non esenti",
xlab="Bilirubina diretta", ylab="Valore numerico", col.main="navy",
medcol="green3", font.lab=2,
ylim=c(0,20))
dev.off()
png(file="Boxplot TSH per esenzione.png", res = 200, width=10, height=6,
units = "in")
par(mfrow=c(1,2))
boxplot(VALORE_NUMERICO[DESCR_risultato=="TSH" & ESENTI=='si'],
main="Distribuzione valori \ndei pazienti esenti", xlab="TSH",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2,
ylim=c(0,43))
boxplot(VALORE_NUMERICO[DESCR_risultato=="TSH" & ESENTI=='no'],
main="Distribuzione valori\ndei pazienti non esenti", xlab="TSH",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2,
ylim=c(0,43))
```

```
dev.off()
png(file="Boxplot Potassio per esenzione.png", res = 200, width=10,
height=6, units = "in")
par(mfrow=c(1,2))
boxplot(VALORE_NUMERICO[DESCR_risultato=="Potassio" & ESENTI=='si'],
main="Distribuzione valori \ndei pazienti esenti", xlab="Potassio",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2)
boxplot(VALORE_NUMERICO[DESCR_risultato=="Potassio" & ESENTI=='no'],
main="Distribuzione valori\ndei pazienti non esenti", xlab="Potassio",
ylab="Valore numerico", col.main="navy", medcol="green3", font.lab=2)
dev.off()

model2 <- glm(FLG_NORMALE_PATOL ~ PZ_ETA_ACC + PZ_SESSO + PZ_CMN_DOMICILIO +
PZ_CMN_NASCITA + PZ_CMN_RESIDENZA + ESENTI, family=binomial)
summary(model2)

confint(model2)
exp(coef(model2))
exp(confint(model2))
1/exp(coef(model2))
1/exp(confint(model2))
1-pchisq(model2$null.deviance, model2$df.null)
1-pchisq(model2$deviance, model2$df.residual)

detach(datiBern)
```

# Conclusione e sviluppi futuri

La struttura di questa tesi rappresenta il risultato del lavoro di questi cinque mesi trascorsi in azienda e che mi hanno portato alle seguenti conclusioni.

Per quanto riguarda il porting dei cruscotti da SpagoBI a Knowage, si è riusciti ad ottenere un'ottimizzazione di questi; per alcuni, ad esempio, è stato diminuito il numero di interrogazioni necessarie per la creazione del cruscotto. Il passo successivo a questo lavoro sarebbe quello di creare un ambiente Knowage apposito per il progetto Pagoda e realizzare quindi il porting di tutto il progetto su questo ambiente.

Riguardo all'analisi avanzata con R, in particolare per il primo problema esposto dal cliente, si è giunti a una conclusione solo preliminare. Questa consiste nell'affermare che è corretto mantenere le stesse soglie nel corso del tempo nonostante il target popolazione sia evoluto e cambiato. Tuttavia, per avere un riscontro più solido, occorrerebbe avere a disposizione un orizzonte temporale molto più ampio rispetto a quello che è stato considerato: anziché su 5 anni sarebbe interessante analizzare l'evoluzione su 50 anni. Un secondo possibile sviluppo di questa analisi sarebbe quello di vedere se c'è una correlazione tra le variazioni dei risultati nel tempo e l'evoluzione della popolazione. Per fare questo occorrerebbe però avere più informazioni sui pazienti, come l'indicazione se il paziente sia immigrato o meno.

Dopo queste prime analisi, l'attenzione è stata spostata sul secondo problema esposto. Si è arrivati a concludere che la popolazione dei pazienti esenti da ticket e quella dei pazienti non esenti non sono omogenee e, quindi, l'esenzione è correlata alla patologicità del risultato. Tuttavia, le possibilità che il

valore dell'esame sia patologico se si è esenti sono solo 1.4 volte le possibilità che sia patologico se non si è esenti; non si ha quindi un valore molto alto (è di poco superiore a 1) tale da poter affermare che il fatto di essere esenti dia la certezza di essere patologici. Sarebbe poi interessante approfondire quale tipologia di esenzione è più soggetta ad essere associata a valori patologici. Per fare questo si potrebbe, ad esempio, effettuare un'analisi tramite cluster. Infine, si è trovato che anche le variabili età e sesso del paziente influiscono sulla patolgicità del risultato dell'analisi.

# Bibliografia


[1] http://eng.it/soluzioni/tecnologie/dettaglio-progetto.dot?inode=89f721d7-dc55-473e-b0e4-bdbe80eebebd&catTecnoId=12c57bc1-7b74-4539-9270-35a0a8b8b142 + 0brochure

[2] http://www.eng.it

[3] https://www.softwareadvice.com/bi

[4] Geoff Hoppe, *Top 17 Free and Open Source Business Intelligence Software*, 26 settembre 2017 https://blog.capterra.com/top-8-free-and-open-source-business-intelligence-software/

[5] Mauro Gasparini, *Modelli probabilistici e statistici - con temi d'esame seconda edizione*, C.L.U.T. Editrice, (2014)

[6] Presentazione progetto Pagoda - Kick-off 2017

[7] Manuale utente Pagoda, Marzo 2017 - V.2

[8] Setti M., Carra D., Sarti M., Trenti T., Cecoli S., *Dal rumore dei dati al data mining*

[9] https://www.knowage-suite.com/site/home/

[10] Knowage Community Edition Manual http://download.forge.ow2.org/knowage/Knowage_6.x_CE_Manual.pdf

[11] https://www.knowage-suite.com/site/wp-content/uploads/2017/06/knowage-brochure-2017.pdf

[12] http://www.dima.unige.it/rogantin/ls_stat/2_BREVE.pdf

[13] William S. Cleveland, *Visualizing Data*, At&T Bell Laboratories, Murray Hill, New Jersey, (1993)

[14] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer, (2013)

[15] https://en.wikipedia.org/wiki/Fisher%27s_exact_test

[16] http://dctf.uniroma1.it/galenotech/campion5.htm

[17] http://www.quadernodiepidemiologia.it/epi/assoc/chi_qua.htm

[18] Levine, Krehbiel, Berenson, *Statistica II ed.*, Capitolo 11, Apogeo, (2006)
http://static.gest.unipd.it/ livio/PDF/Test%20Chi-Quadrato.pdf

[19] Appunti del corso magistrale di "Modelli statistici"

[20] Deborah Ascolese, *I Benefici Della Business Intelligence In Sanità*, 17 dicembre 2016
https://www.gipo.it/business-intelligence-in-sanita/

[21] Luca Gastaldi, *La Business Intelligence che cambia la Sanità*, 26 luglio 2013
https://www.agendadigitale.eu/cittadinanza-digitale/la-business-intelligence-che-cambia-la-sanita/

[22] Slide del corso magistrale di "Business Intelligence"

[23] Alessandro Rezzani, *Business Intelligence*, Apogeo, (2012)
http://www.apogeonline.com/2012/libri/9788850331055/ebook/pdf/3105_capitolo1Estratto.pdf

[24] Giampiero Carli Ballola, *Business Intelligence: la linea di evoluzione della specie*, 19 aprile 2010
https://www.zerounoweb.it/analytics/business-intelligence-la-linea-di-evoluzione-della-specie/

[25] http://bi.gruppocdm.it/levoluzione-senza-fine-della-business-intelligence/

[26] http://www.dwreview.com/DW_Overview.html