SCUOLA DI DOTTORATO

UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Dipartimento di / Department of

MEDICINA E CHIRURGIA

Dottorato di Ricerca in / PhD program SANITÀ PUBBLICA Ciclo / Cycle XXXVI

Curriculum in (se presente / if it is) BIOSTATISTICA ED EPIDEMIOLOGIA

INFERENZA SU DATI PESATI: APPLICAZIONE E SIMULAZIONI IN UN CASO DI STUDIO

Cognome / Surname PARODI Nome / Name ANDREA

Matricola / Registration number 079343

Tutore / Tutor: ANTOLINI LAURA

Cotutore / Co-tutor: BERNASCONI DAVIDE

Supervisor: ANTOLINI LAURA

Coordinatore / Coordinator: BADANO LUIGI

ANNO ACCADEMICO / ACADEMIC YEAR 2023/2024

Table of Contents

Introduzione

Il presente lavoro di tesi si concentra sull'applicazione e la valutazione critica di metodologie di weighting, con particolare attenzione al confronto tra due fonti di dati con Individual Patient Data (IPD).

L'obiettivo principale della ricerca è stato indagare le tecniche di inferenza nel contesto di dati pesati, attraverso simulazioni basate su un caso di studio reale. In particolare, l'attenzione è stata rivolta a capire quale tecnica permettesse di ottenere un risultato accurato, senza inflazione dell'errore di tipo I, massimizzando la potenza e minimizzando la numerosità campionaria.

Contesto clinico

Mi è stato chiesto da una società che si occupa di dispositivi medici per neuromodulazione e chirurgia cardiovascolare di disegnare uno studio prospettico, multicentrico, osservazionale, a gruppi paralleli, per valutare l'efficacia e la sicurezza della terapia Vagus Nerve Stimulation (VNS) confrontata con lo Standard of Care farmacologico. Il VNS è un device impiantabile per il trattamento dell'epilessia farmaco-resistente [1].

L'epilessia è uno dei disordini neurologici più diffusi a livello globale, con un impatto significativo sulla qualità della vita dei pazienti. Nonostante i progressi nella farmacoterapia, circa il 30% dei pazienti affetti da epilessia sviluppa una forma resistente ai farmaci, nota come drug-resistant epilepsy (DRE). Secondo la definizione proposta dall'International League Against Epilepsy (ILAE), la DRE si manifesta quando due appropriati regimi terapeutici di farmaci antiepilettici (FAE), scelti in base al tipo di crisi e utilizzati in monoterapia o in combinazione, non riescono a controllare le crisi epilettiche del paziente [2].

La resistenza ai farmaci antiepilettici rappresenta una sfida clinica complessa e multifattoriale. Le cause della DRE non sono ancora completamente comprese, ma numerosi fattori sono stati ipotizzati, tra cui alterazioni farmacocinetiche, meccanismi genetici e strutturali, e modifiche nella plasticità neuronale. Inoltre, i pazienti con DRE presentano un rischio maggiore di comorbidità psichiatriche, cognitive e sociali, il che rende la gestione della malattia particolarmente difficoltosa [3].

Nel corso degli anni, sono state esplorate diverse strategie terapeutiche per il trattamento della DRE, tra cui la chirurgia dell'epilessia, la stimolazione cerebrale profonda (DBS) e la dieta chetogenica. Sebbene questi approcci possano migliorare il controllo delle crisi in alcuni pazienti, non esiste ancora una soluzione universalmente efficace. La ricerca continua a focalizzarsi sull'identificazione di nuovi biomarcatori prognostici e di terapie innovative per migliorare la gestione della DRE, tra cui la stimolazione del nervo vago (VNS).

L'endpoint primario dello studio era la durata massima della risposta al trattamento, quantificata in mensilità.

La durata dello studio prefissata era di 36 mesi per ogni paziente, e nelle 36 visite mensili prefissate si sarebbero raccolti gli episodi epilettici occorsi dalla visita precedente. Il massimo di mesi consecutivi senza episodi epilettici rappresenta l'endpoint primario in studio. In tabella 1 sono

rappresentati 4 esempi per capire meglio la definizione dell'endpoint primario (0 = nessun episodio, 1 = almeno 1 episodio, in verde il periodo selezionato).

Sebbene io ritenessi che la costruzione di questo endpoint non fosse ottimale, non è stato purtroppo possibile un dialogo con l'azienda in questi termini, avendo utilizzato loro questo endpoint in diversi studi clinici precedenti e volendo fortemente proseguire con la stessa modalità.

L'endpoint in sé non riesce a catturare la durata del trattamento fino all'evento di interesse e non è in grado di valutare un possibile andamento temporale dell'efficacia della terapia in esame né tantomeno di un suo possibile mantenimento. Avrei ritenuto un time-to-event o delle misure ripetute più adeguate a testare l'efficacia del trattamento in relazione alla sua durata.

Tabella 1 - Esempi endpoint primario

Mesi	Paziente 1	Patiente 2	Paziente 3	Paziente 4
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	0	0	0
5	1	0	1	0
6	1	0	1	0
7	1	0	1	0
8	1	0	1	0
9	1	0	1	0
10	1	0	0	0
11	1	0	1	0
12	1	1	0	0
13	1	0	0	0
14	1	0	0	0
15	1	0	1	0
16	1	0	1	0
17	1	0	1	0
18	1	1	1	0
19	1	0	1	0
20	1	0	1	0
21	1	0	1	0
22	1	0	1	0
23	1	0	1	0
24	1	0	1	0
25	1	0	1	0
26	1	0	1	0
27	1	0	0	0
28	1	0	0	0
29	1	0	1	0
30	1	0	0	0
31	1	0	0	0
32	1	0	0	0
33	1	1	1	0
34	1	0	0	0
35	1	0	0	0
36	1	0	1	0
Risultato	0	14	4	36

La scelta del trattamento era decisa dal medico e dal paziente prima dell'entrata nello studio.

Nota su un'esperienza passata.

L-MIND è uno studio di fase 2 a braccio singolo e in aperto, che investiga la combinazione di tafasitamab e lenalidomide in pazienti con linfoma diffuso a grandi cellule B recidivante o refrattario (r/r DLBCL) dopo un massimo di due linee di terapia precedenti, inclusa una terapia mirata anti-CD20 (ad esempio, rituximab), che non sono idonei per la chemioterapia ad alte dosi e il successivo trapianto autologo di cellule staminali. L'endpoint primario dello studio è il tasso di risposta obiettiva (ORR). Le misure di esito secondarie includono la durata della risposta (DoR), la sopravvivenza libera da progressione (PFS) e la sopravvivenza globale (OS). Nel maggio 2019, lo studio ha raggiunto il suo completamento primario. I dati dell'analisi primaria hanno incluso 80 pazienti arruolati nello studio che avevano ricevuto tafasitamab e lenalidomide e che erano stati seguiti secondo il protocollo per almeno un anno [4]. Il 09 Novembre 2018 ho preso parte ad un Type C meeting con l'FDA per discutere l'utilizzo di Real World Data per fornire evidenze sull'effetto additivo del tafasitamab rispetto al solo utilizzo di lenalidomide.

Lo studio Re-MIND, uno studio osservazionale retrospettivo, è stato progettato per isolare il contributo di tafasitamab nella combinazione con lenalidomide e per dimostrare l'effetto combinatorio. Lo studio confronta i dati di risposta del mondo reale di pazienti con DLBCL recidivante o refrattario che hanno ricevuto la monoterapia con lenalidomide con i risultati di efficacia della combinazione tafasitamab-lenalidomide, come investigato nello studio L-MIND. Re-MIND ha raccolto i dati di efficacia da 490 pazienti r/r DLBCL negli Stati Uniti e nell'UE [5].

Durante la discussione nel Type C meeting, ho proposto di rendere confrontabili i risultati dei due studi utilizzando delle tecniche di matching o di weighting [6]. La preferenza dell'FDA è ricaduta sul matching, ritenendo l'inferenza su dati non pesati più affidabile, senza aggiungere ulteriori spiegazioni né referenze.

Quella risposta ha suscitato la mia curiosità, sfociata poi in questo progetto di ricerca, con l'obiettivo di capire se le tecniche di weighting potessero essere ritenute affidabili quanto quelle di matching.

In seguito, sono stati identificati 76 pazienti idonei in Re-MIND e appaiati 1:1 a 76 degli 80 pazienti di L-MIND in base a importanti caratteristiche di base. I tassi di risposta obiettiva (ORR) sono stati convalidati in base a questo sottogruppo di 76 pazienti in Re-MIND e L-MIND, rispettivamente. L'endpoint primario di Re-MIND è stato raggiunto e mostra un ORR migliore e statisticamente significativo della combinazione tafasitamab/lenalidomide rispetto alla monoterapia con lenalidomide.

Le approvazioni iniziali negli Stati Uniti e nell'Unione Europea del tafasitamab rappresentano un precedente, poiché è la prima volta che l'approvazione di una nuova terapia combinata è stata concessa sulla base di uno studio pivotal a braccio singolo (SAT). L'appaiamento con RWD ha aiutato a scomporre il contributo dei singoli agenti [7].

Aspetti statistici

In assenza di randomizzazione è stata quindi l'occasione per approfondire le tecniche di weighting, ricadenti nel più ampio argomento dell'Indirect Treatment Comparison (ITC), un insieme di tecniche analitiche utilizzate nel campo della ricerca sanitaria, per confrontare l'efficacia e la sicurezza di diversi trattamenti quando i dati derivano da studi clinici non randomizzati o da fonti eterogenee.

La prima richiesta che mi è stata fatta per finalizzare il disegno di studio è stato il calcolo della numerosità campionaria. Per definire la numerosità campionaria, è però necessario avere chiara la misura di effetto da utilizzare per valutare la differenza tra i due gruppi di trattamento nell'endpoint primario e stabilire come fare inferenza sulla differenza tra i gruppi. Da qui la necessità di una simulazione per capire quale metodo di inferenza fosse il migliore.

Letteratura

Gli studi di Austin hanno rappresentato la base statistica per ideare le simulazioni presenti in questo lavoro.

Il propensity score è la probabilità di assegnazione al trattamento condizionata alle caratteristiche basali osservate. Esso consente di progettare e analizzare uno studio non randomizzato in modo tale da imitare alcune delle caratteristiche specifiche di un trial controllato randomizzato. In particolare, il propensity score è uno score di bilanciamento: condizionatamente al propensity score, la distribuzione delle covariate basali osservate sarà simile tra i soggetti trattati e non trattati [6].

Inizialmente erano tre i metodi basati sul propensity score prevalentemente utilizzati: aggiustamento utilizzando il propensity score come covariata nei modelli, stratificazione basata sul propensity score e matching [8, 9]. La ricerca si è concentrata nei primi anni nel constatare come il matching fosse superiore alle altre tecniche. Austin ha dimostrato che la precisione delle stime marginali degli odds ratio era migliore col matching rispetto agli altri due metodi, ottenendo inoltre il più basso mean square error [10]. Analogo risultato è stato ottenuto per quanto riguardava i rischi relativi [11].

Inizialmente, Austin suggeriva di trattare i dati appaiati usando metodi per dati dipendenti, trattando le coppie come se fossero realmente appaiate [12, 13], ma in seguito alcuni autori hanno contestato questa scelta [14]. A seguito di indagini condotte con simulazioni di Montecarlo, Austin dimostrò come i metodi per dati appaiati portassero a stime intervallari più precise [15, 16].

Nella preziosa bibliografia di Austin, di rilevante importanza sono le tecniche diagnostiche per valutare il bilanciamento delle covariate al baseline [17]. Basandosi su queste tecniche diagnostiche, nel 2009 Austin arrivò ad affermare come le tecniche di stratificazione e l'utilizzo del propensity score come covariata nei modelli potessero essere abbandonate a vantaggio delle tecniche di matching (come già noto) e per la prima volta delle tecniche di weighting [18]. La prima tecnica di weighting proposta da Austin è l'Inverse Probability of Treatment Weighting, descritta di seguito:

Sia T l'assegnazione del trattamento (dove T=1 indica la presenza del trattamento e T=0 l'assenza del trattamento). Inoltre, sia Z il propensity score, ovvero la probabilità che un soggetto riceva il trattamento di interesse condizionata alle covariate basali osservate. L'inverse probability of treatment è definito come:

$$\frac{T}{Z} + \frac{1-T}{1-Z}$$

Per ogni soggetto, questo valore corrisponde all'inverso della probabilità di ricevere il trattamento effettivamente assegnato.

Austin ha, sempre nel 2009, individuato come limite per standardized difference post matching (o post IPTW) il 10% per dichiarare le differenze nelle variabili al baseline come negligibili [19].

Studi focalizzati su outcome binari con risk difference come misura di associazione, hanno in seguito mostrato come lo stimatore IPTW proposto da Lunceford and Davidian sia il più accurato e preciso [20, 21]. Negli ultimi anni, l'attenzione è stata però posta sui metodi di stima della varianza dell'effetto del trattamento, che può portare ad una stima distorta dello standard error [22].

I metodi per la stima della varianza quando si utilizzano tecniche basate sul propensity score hanno ricevuto scarsa attenzione nella letteratura metodologica. Nel contesto del propensity score matching, studi basati su simulazioni Monte Carlo hanno dimostrato che la stima della varianza deve tenere conto della natura matched del campione e che stime naïve della varianza sono inadeguate [15, 16, 23, 24].

Quando si utilizza il propensity score matching con reinserimento, Abadie e Imbens hanno osservato che il bootstrapping per stimare gli errori standard è inappropriato [25]. Tuttavia, in caso di matching senza reinserimento (without replacement), Austin e Small hanno riscontrato che il bootstrapping funziona bene [26].

Per quanto riguarda l'IPTW (Inverse Probability of Treatment Weighting), Lunceford e Davidian hanno sviluppato stimatori formali della varianza quando si stima l'effetto lineare del trattamento (ad esempio, differenze di medie o proporzioni) [27]. Joffe et al., invece, hanno raccomandato l'uso di uno stimatore robusto della varianza quando si adatta un modello di regressione a un campione ponderato con pesi IPTW [28].

In un lavoro di Xu et al. [29], attraverso simulazioni Monte Carlo, sono stati confrontati tre metodi di stima per dati di sopravvivenza:

- Pesi IPTW-ATE convenzionali con stimatore naïve della varianza.
- Pesi IPTW-ATE stabilizzati con stimatore naïve della varianza.
- Pesi IPTW-ATE convenzionali con stimatore robusto della varianza.

Il loro obiettivo principale era valutare la dimensione del campione ponderato e l'errore di tipo I. I risultati hanno mostrato che:

- L'uso di pesi IPTW-ATE convenzionali produceva un campione ponderato circa doppio rispetto a quello originale.
- L'uso di pesi stabilizzati manteneva una dimensione simile al campione non ponderato.
- I pesi convenzionali con stimatore naïve portavano a una sovrastima dell'errore di tipo I.
- I pesi stabilizzati con stimatore naïve producevano tassi di errore di tipo I più accurati.
- L'uso di pesi stabilizzati con stimatore robusto riduceva l'errore di tipo I al di sotto del 5%.

Questi risultati sono in linea con il paper di Austin et al. [22], dove lo stimatore naïve con pesi convenzionali sottostimava la varianza e inflazionava l'errore di tipo I di dati di sopravvivenza. Tuttavia, Autin et al. hanno anche osservato che i pesi stabilizzati con stimatore naïve introducevano un certo bias, seppur minore rispetto ai pesi convenzionali. Queste differenze potrebbero dipendere dalla natura dell'outcome (binario vs. tempo-evento) o dal disegno simulato (Xu et al. hanno usato due covariate, mentre Austin et al. ne hanno incluse 10, miste tra continue e dicotomiche). Austin et al. hanno quindi concluso raccomandando l'uso del bootstrapping per stimare errori standard e intervalli di confidenza quando si applica un modello di Cox proporzionale a campioni ponderati con IPTW.

Uno studio di Williamson et al. [30] fornisce una possibile spiegazione per le prestazioni non ottimali dello stimatore robusto. Analizzando trial randomizzati con IPTW, hanno sviluppato uno stimatore della varianza per dati continui e binari che tiene conto del fatto che i pesi sono stimati e non noti con certezza. Hernan et al. [31] avevano già osservato che la ponderazione introduce una correlazione

intra-soggetto, giustificando l'uso di stimatori robusti. Tuttavia, questi ultimi non considerano l'incertezza nella stima del propensity score.

Al contrario, il bootstrapping incorpora questa variabilità, ricalcolando il propensity score in ogni campione estratto.

Siccome la richiesta che mi è stata fatta riguardava una situazione differente da quella trattata nei lavori citati in precedenza, ho ritenuto utile procedere alle simulazioni per chiarire il metodo da scegliere per fare inferenza in questo specifico caso.

In particolare, la variabile risposta era discreta (assume valori discreti da 0 a 36), con distribuzione non assimilabile ad alcuna nota, ed era inoltre mia intenzione utilizzare un metodo di weighting alternativo all'IPTW che, come spiegato in dettaglio nei paragrafi successivi, si adatta alla perfezione alla richiesta ricevuta.

Metodi

Simulazione delle popolazioni

Due popolazioni da un milione di soggetti ciascuna, una per gli utilizzatori di VNS, l'altra per gli utilizzatori di SoC (Standard of Care), sono state simulate in base ai seguenti input clinici.

Tabella 2 - Parametri al baseline

Parametri	VNS	SoC
Decadimento cognitivo / difficoltà d'apprendimento	60%	40%
Numero di SoC fallite	Media (SD) 6.8 (3.06) Min 3, Max 9	Media (SD) 8.1 (4.10) Min 3, Max 9
Durata dell'epilessia (anni)	Media (SD) 14.4 (12.74) Min 0, Max 20	Media (SD) 24.9 (13.40) Min 0, Max 20
Età	Media (SD) 23.9 (16.55) Min 4, Max 70	Media (SD) 36 (18.0) Min 4, Max 70
Tipi di convulsione	Focale 70% Generalizzata 30%	Focale 80% Generalizzata 20%
Tasso di convulsioni al mese (al basale)	[2, 10): 28% [10, 100): 50% [100, 1000): 22%	[2, 10): 28% [10, 100): 50% [100, 1000): 22%

La durata massima della risposta al trattamento è stata simulata con probabilità differenziate in base alle caratteristiche al baseline e agli episodi epilettici dei soggetti nei tre mesi precedenti all'entrata nello studio (Appendice A), ottenendo:

Tabella 3 - Durata massima della risposta al trattamento (su popolazione osservata)

	VNS	SoC
Media (SD)	14.87 (12.988)	11.72 (12.204)
Mediana	14	7
Q1 – Q3	1 – 27	0 – 20

Dai dati riportati in
Tabella 3 - Durata massima della risposta al trattamento, ricaviamo una differenza media tra le due popolazioni di 3.15 mesi. Tale differenza non riflette però la vera misura di effetto presente tra le due popolazioni, in quanto distorta dalle diverse caratteristiche al basale delle due popolazioni.
Purtroppo, in base ai complessi input clinici sui quali sono state simulate le popolazioni, non è possibile sapere la vera misura di effetto del trattamento a priori e non è possibile partire da un'ipotesi predefinita per il confronto.
Tuttavia, applicando dei pesi (Inverse Probability Weights o Overlap Weights, trattati nelle seguenti sezioni di questa tesi) ai pazienti o calcolando delle stime marginali parametriche si può evincere come la vera media tra le differenze nella durata massima del trattamento sia minore di 3.15 (Tabella 4 - Durata massima della risposta al trattamento):

Tabella 4 - Durata massima della risposta al trattamento (su popolazione potenzialmente osservabile)

Media calcolata con:	VNS	SoC	Differenza tra medie
IPW	14.10	11.99	2.11
Overlap Weights	14.47	12.10	2.37
Stima marginale (distribuzione normale)	16.39	13.99	2.40
Stima marginale (distribuzione poisson)	16.84	14.24	2.60
Stima marginale (distribuzione binomiale negativa)	17.30	14.39	2.91

La figura 1 mostra la distribuzione del primary endpoint.

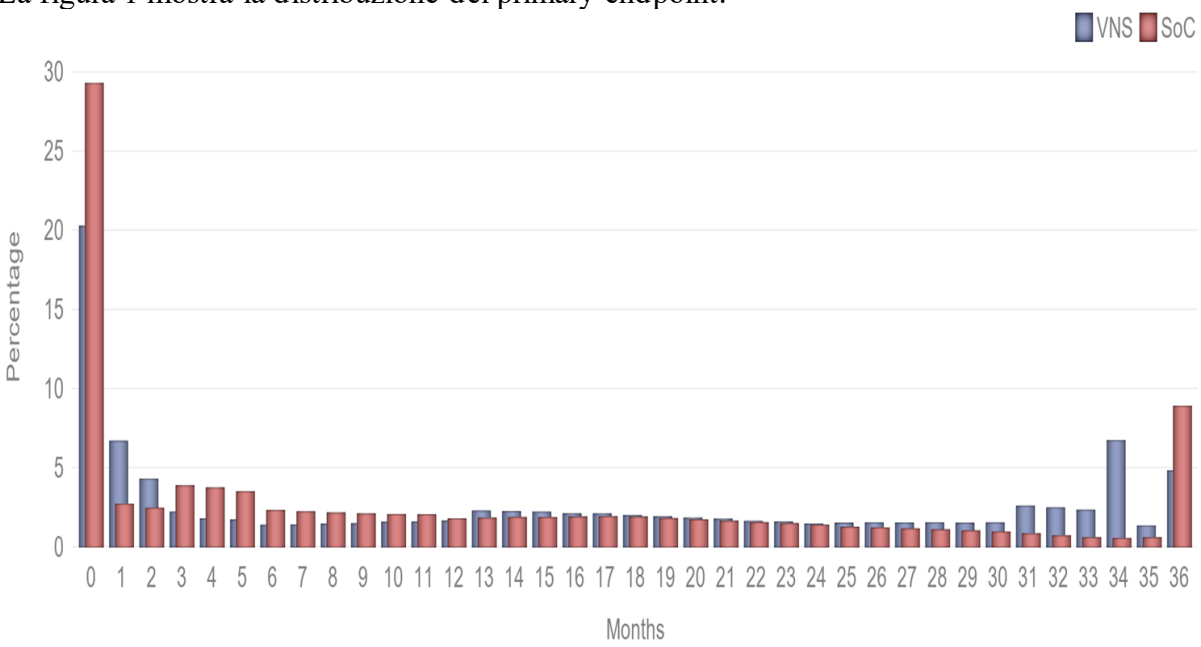

Figura 1 - Distribuzione della durata massima della risposta al trattamento

Come si può notare dal grafico, la distribuzione del primary endpoint non segue nessuna distribuzione nota. Tuttavia, nella pianificazione degli studi clinici è pratica comune ipotizzare una distribuzione per la variabile di risposta senza avere certezza a priori che tale assunzione verrà rispettata. Questo scenario riflette fedelmente le difficoltà di applicazione nel mondo reale, dove spesso non si può prevedere con esattezza la forma della distribuzione dei dati.

Overlap Weights

La tecnica degli Overlap Weights sarà utilizzata per minimizzare le differenze al baseline tra le due popolazioni simulate e il primary endpoint sarà analizzato con diverse tecniche statistiche per valutare quale sia la migliore in termini di sample size necessario e inflazione dell'errore di Tipo 1.
Gli Overlap Weights sono una metodologia statistica sviluppata da Li et al [32] utilizzata per il bilanciamento delle covariate in studi osservazionali o studi senza randomizzazione. Questa tecnica è stata proposta per migliorare l'equilibrio delle covariate rispetto ai metodi tradizionali di weighting. Si è scelto di utilizzare questo tipo di pesi in quanto si adatta perfettamente allo studio in oggetto.
I diversi approcci di utilizzo dei pesi presenti in letteratura possono portare a 4 diverse stime di misure di effetto:

1. ATT (Average Treatment effect on the Treated)
2. ATC (Average Treatment effect on the Controls)
3. ATE (Average Treatment effect)
4. ATO (Average Treatment effect for the Overlap Population)

Le stime ATT e ATC rappresentano l'effetto medio di un trattamento rispettivamente sui soggetti che hanno ricevuto il trattamento e sui soggetti che hanno ricevuto il controllo.
Le stime ATE (calcolate con pesi definiti Inverse Probability Weights IPW) rappresentano l'effetto medio di un trattamento su tutti i soggetti, trattati e non.
Le stime ATO (calcolate con gli Overlap Weights) infine, rappresentano l'effetto medio di un trattamento sui soggetti che avrebbero potuto prendere sia il trattamento che il controllo, definita overlap population.
Le stime ATE e ATT sono ampiamente accettate, soprattutto perché rappresentano effetti su popolazioni target facili da interpretare. Tuttavia, la scelta automatica di queste stime è spesso discutibile nella pratica. Innanzitutto, l'ATE (Average Treatment Effect) e l'ATT (Average Treatment effect on the Treated) possono corrispondere all'effetto di un intervento irrealizzabile. Ad esempio, l'ATE corrisponde all'effetto di passare ogni unità nella popolazione dello studio da un trattamento all'altro; un cambiamento così completo del trattamento è raramente concepibile negli studi clinici, dove potrebbe essere noto che il trattamento è dannoso per pazienti con certe caratteristiche. Inoltre, come spiegato da Rosenbaum [33], spesso i dati disponibili non rappresentano una popolazione naturale, e quindi non vi è una ragione convincente per stimare l'effetto del trattamento su tutte le persone della popolazione.
In tali casi, applicare i pesi ATE o ATT al campione osservato potrebbe non fornire una stima dell'effetto medio del trattamento nella popolazione target scientificamente appropriata. In secondo luogo, i ricercatori spesso escludono alcune unità dall'analisi finale, ad esempio unità con pesi estremamente elevati. Il campione rimanente può rappresentare solo una sottopopolazione della popolazione target originaria, e questa sottopopolazione può variare da studio a studio. In terzo luogo, con ATE e ATT, probabilità estreme (vicine a 0 o 1) introducono pesi inversi estremi, con conseguenze sfavorevoli in campioni finiti, come scarso bilanciamento e ampia varianza [34]. Infine, nell'inferenza causale è spesso ritenuto prioritario il bilanciamento delle covariate tra i gruppi di trattamento rispetto alla generalizzabilità verso una popolazione target riconoscibile, come dimostrato dalla vasta evidenza sperimentale basata sulla randomizzazione del trattamento all'interno di un campione non selezionato casualmente (in molti casi, di convenienza).
Gli ATO, a differenza di ATE e ATT sono limitati tra 0 e 1, pertanto le distribuzioni ponderate risultano sempre integrabili. Gli ATO minimizzano inoltre la varianza asintotica della stima non parametrica dell'effetto medio ponderato del trattamento rispetto agli ATT e ATE.
Gli ATO stimati da un modello logistico possiedono inoltre una proprietà utile: garantiscono un bilanciamento esatto tra i gruppi nelle medie di ciascuna covariata inclusa nel modello.
Al di là della robustezza del metodo descritto da Li et al, lo scopo dello studio è proprio quello di stimare la misura di effetto in una popolazione che avrebbe potuto scegliere sia il trattamento invasivo con VNS che il trattamento farmacologico con lo Standard of Care (SoC).

Descrizione della Tecnica

1. **Propensity Score**: Calcolare il propensity score *e(x)*, che rappresenta la probabilità di ricevere il trattamento VNS data una serie di covariate *x*.

Codice SAS:

```
proc logistic data=<dataset> noprint;
class <list of covariates>;
model trtgroup=<list of covariates>;
output out=ps predicted=ps;
run;
```

2. **Calcolo dei Pesi**: Gli overlap weights vengono calcolati come:

$$VNS: w_i = 1 - e(x)$$

$$SoC: w_i = e(x)$$

Questo peso attribuisce maggiore importanza agli individui con una probabilità intermedia di ricevere il trattamento, riducendo il peso di coloro che hanno una probabilità molto alta o molto bassa.

3. **Bilanciamento delle Covariate**: L'uso degli overlap weights tende a bilanciare le covariate tra i gruppi trattati e di controllo, migliorando la comparabilità e riducendo il bias.
4. **Stima degli Effetti del Trattamento**: Gli effetti del trattamento possono essere stimati utilizzando modelli ponderati dagli overlap weights, migliorando così l'accuratezza delle stime.

Vantaggi

- **Bilanciamento Migliorato**: Gli overlap weights tendono a bilanciare le covariate in modo più efficace rispetto agli inverse probability weights. Questo è dovuto al fatto che gli overlap weights concentrano l'analisi sugli individui con una probabilità intermedia di ricevere il trattamento, dove vi è una maggiore sovrapposizione delle covariate tra i gruppi di trattamento e controllo.
- **Riduzione del Bias**: Riducendo il peso degli individui con probabilità estreme, si riduce il potenziale bias nella stima degli effetti del trattamento.
- **Robustezza**: La tecnica è robusta a diverse specificazioni del modello di propensity score.

Li et al. presentano risultati empirici che mostrano come gli overlap weights migliorano il bilanciamento delle covariate e riducono il bias in diversi scenari simulati e applicazioni reali. Questi risultati supportano la superiorità degli overlap weights in termini di bilanciamento delle covariate e robustezza delle stime degli effetti del trattamento.

Dettagli della simulazione

1. Un numero di soggetti pari a V (V iniziale = 10 soggetti) è stato estratto casualmente senza sostituzione dalla popolazione VNS per 10.000 volte.
   Un numero di soggetti pari a S (con S = V x 2) è stato estratto casualmente senza sostituzione dalla popolazione SoC per 10.000 volte.
   Questo passaggio ha creato 10.000 dataset per i soggetti VNS denominati VNS0001-VNS10000 così come 10.000 dataset per i soggetti SoC denominati SOC0001-SOC10000.

   Codice SAS:

```
proc surveyselect data=VNS out=outboot seed=1
method=urs samprate=V/1000000 outhits noprint
rep=10000;
run;
```

2. Per ogni coppia di dataset VNS0001-SOC0001, VNS0002-SOC0002 … VNS10000-SOC10000, è stata eseguita una regressione logistica, utilizzando l'assegnazione del trattamento VNS come evento, e le 6 variabili della Tabella 1 come covariate. La probabilità predetta di essere assegnato al gruppo VNS è chiamata Propensity Score (PS). Il peso per ciascun soggetto è stato calcolato in base al PS: per i soggetti SoC, l'Overlap Weight (OLW) è pari al PS, mentre per quelli VNS è pari a 1 - PS.

   Codice SAS:

```
proc logistic data=<dataset> noprint;
class <list of covariates>;
model trtgroup=<list of covariates>;
output out=ps predicted=ps;
run;
```

3. Per ogni coppia di dataset VNS0001-SOC0001, VNS0002-SOC0002 … VNS10000-SOC10000, è stata calcolata la differenza ponderata nella media della durata della risposta massima nei due gruppi di trattamento, insieme ai suoi intervalli di confidenza al 95% calcolati con il metodo bootstrap (estraendo casualmente i soggetti con sostituzione). La media ponderata è stata calcolata usando la seguente formula:

$$\frac{\sum_{i=1}^{N} w_i \times x_i}{\sum_{i=1}^{N} w_i}$$

   dove N è il numero di soggetti per gruppo di trattamento (pari quindi a V o S), w_i è il peso calcolato al punto 2 per il soggetto i e x_i la durata della risposta massima per il soggetto i.

Codice SAS:

```
proc surveyselect data=outboot_<n> out=btr_<n>(drop=NumberHits
ExpectedHits SamplingWeight)
seed=1 method=urs samprate=1 outhits noprint rep=10000;
strata TrtGroup;
run;

proc means data=btr_<n> noprint;
by Replicate TrtGroup;
var dur36;
output out=ciest_<n>(drop=_TYPE_ _FREQ_) mean=dur36;
weight olw;
run;
```

4. Se il limite inferiore dell'intervallo di confidenza n-esimo è > 0, allora l'n-esima simulazione è stata considerata un successo. La potenza dell'analisi con V soggetti VNS e S soggetti SoC è stata calcolata dividendo il numero di successi per le 10.000 simulazioni effettuate.

I passaggi 1-4 sono stati ripetuti con numeri crescenti di V e S fino a quando la potenza non è stata ≥ 80%. Il rapporto V:S è stato impostato a 1:2, come descritto nel primo step, in quanto il numero di soggetti disposti all'impianto è solitamente circa la metà di quelli che intendono iniziare/continuare una terapia medica.
Il sample size necessario per mostrare una superiorità significativa con test unilaterale, livello di significatività alpha = 0.025 e con l'80% di potenza è stato calcolato oltre che con l'approccio non parametrico appena descritto, anche con i seguenti approcci:

Tabella 5 - Approcci parametrici

Distribuzione assunta	Calcolo degli intervalli di confidenza
Normale	Wald
Normale	Sandwich
Log-normale	Wald
Log-normale	Sandwich
Poisson	Wald
Poisson	Sandwich
Binomiale negativa	Wald
Binomiale negativa	Sandwich

In base alle distribuzioni mostrate in Figura 1, si evince chiaramente che la distribuzione dell'endpoint non è assimilabile ad alcuna delle distribuzioni presenti in Tabella 5. Nonostante ciò si è ritenuto di procedere ugualmente al confronto perché nella pianificazione degli studi clinici è pratica comune ipotizzare una distribuzione per la variabile di risposta senza avere certezza a priori che tale

assunzione verrà rispettata. Questo scenario riflette fedelmente le difficoltà di applicazione nel mondo reale, dove spesso non si può prevedere con esattezza la forma della distribuzione dei dati.
I modelli parametrici rispecchiano inoltre un approccio computazionalmente rapido alla soluzione del problema, mentre il bootstrapping richiede l'estrazione di *n* campioni (in questo specifico caso 10'000) sui quali viene poi calcolata la differenza tra medie, richiedendo quindi delle tempistiche ampiamente più larghe, contrastando con l'esigenza delle aziende farmaceutiche di pubblicare il risultati ottenuti nel più breve tempo possibile, in modo che la nuova terapia sperimentale possa raggiungere i pazienti nel più breve tempo possibile.
Per ciascuna delle distribuzioni assunte, gli intervalli di confidenza sono stati calcolati sia con l'approccio di Wald che con il metodo robusto di Sandwich. Di seguito, vengono descritte in dettaglio le differenze principali tra questi due metodi e le distribuzioni utilizzate.
L'approccio di Wald, comunemente utilizzato, non risulta adeguato quando si applicano pesi di bilanciamento come gli overlap weights. Questo perché l'approccio standard di Wald assume omoschedasticità e indipendenza degli errori, condizioni che possono essere violate nei dati ponderati. Pertanto, per ottenere stime accurate della varianza, è preferibile utilizzare stime robuste come il metodo Sandwich, che correggono l'eteroschedasticità e migliorano l'affidabilità dell'inferenza. Nonostante ciò, l'approccio di Wald per l'inferenza è stato inserito nelle simulazioni per controllare empiricamente i risultati del suo utilizzo in questa specifica situazione.

Approccio di Wald

L'approccio di Wald è uno dei metodi tradizionali per stimare intervalli di confidenza e testare ipotesi sui parametri di un modello. Viene utilizzato con modelli basati su diverse distribuzioni, tra cui quella normale, log-normale, di Poisson e binomiale negativa. Le principali assunzioni di questo approccio sono:

- **Normalità asintotica**: Presuppone che per campioni grandi, le stime dei parametri seguano una distribuzione normale. Questo deriva dal teorema del limite centrale, che giustifica l'uso della normale anche se la distribuzione originale dei dati non lo è.
- **Varianza omoschedastica**: Si assume che la varianza degli errori sia costante in tutte le osservazioni. Questa condizione può essere restrittiva, specialmente per dati di conteggio o in presenza di eteroschedasticità.

Nel metodo di Wald, la matrice di varianza-covarianza gioca un ruolo cruciale per il calcolo degli errori standard e degli intervalli di confidenza. In un modello classico di regressione lineare, la matrice di varianza-covarianza è definita come:

$$V = \hat{\sigma}^2 (X^T X)^{-1}$$

dove:

- $\hat{\sigma}^2$ è la varianza stimata degli errori (assunta costante per tutte le osservazioni),
- X è la matrice delle covariate (che contiene i valori delle variabili indipendenti),
- X^T è la trasposta di X.

Questa matrice rappresenta la struttura della varianza e della covarianza tra gli stimatori dei parametri nel modello. Gli errori standard degli stimatori dei parametri sono ottenuti dalla radice quadrata degli elementi diagonali di questa matrice.
Una volta calcolata la matrice di varianza-covarianza, gli intervalli di confidenza per i parametri del modello sono calcolati utilizzando la seguente formula:

$$\hat{\beta} \pm Z \times SE(\hat{\beta})$$

dove $\hat{\beta}$ è la stima del parametro, Z è il quantile della distribuzione normale standard, e $SE(\hat{\beta})$ è l'errore standard della stima.
Questo approccio è stato applicato a tutte le distribuzioni considerate (Normale, Log-normale, Poisson, Binomiale negativa), ma può risultare inaffidabile quando le assunzioni di base non sono rispettate [35].

Approccio robusto di Sandwich
L'approccio Sandwich è utilizzato per calcolare intervalli di confidenza più robusti quando le assunzioni di omoschedasticità e indipendenza degli errori non sono rispettate. Questo metodo tiene conto di:

- **Eteroschedasticità**: La varianza degli errori può variare tra le osservazioni.
- **Correlazione nei residui**: Gli errori possono non essere indipendenti tra di loro, specialmente in contesti longitudinali o con dati raggruppati.

Il metodo Sandwich utilizza una matrice di varianza-covarianza robusta, che corregge l'errore standard degli stimatori senza modificare la stima dei parametri stessi. Questo lo rende meno sensibile a violazioni delle assunzioni classiche, offrendo quindi stime più affidabili in situazioni in cui l'approccio di Wald può essere fuorviante.
Anche gli intervalli di confidenza ottenuti con il metodo Sandwich sono simmetrici, ma riflettono meglio la variabilità effettiva dei dati, risultando più adeguati quando si riscontrano problemi come l'eteroschedasticità [36].
La matrice di varianza-covarianza rappresenta la struttura della variabilità e delle correlazioni tra gli stimatori dei parametri del modello. In un modello classico, questa matrice viene utilizzata per calcolare gli errori standard degli stimatori e, di conseguenza, gli intervalli di confidenza.
Nel metodo di Sandwich, si utilizza una versione robusta della matrice di varianza-covarianza. La matrice di varianza-covarianza robusta corregge l'errore standard degli stimatori, tenendo conto delle deviazioni dalle assunzioni di eteroschedasticità e correlazione.
La matrice robusta è calcolata come:

$$V_{robusto} = (X^T X)^{-1} X^T \Omega X (X^T X)^{-1}$$

dove:

- X è la matrice delle covariate (ossia le variabili indipendenti del modello),
- Ω è una matrice diagonale con i residui stimati (la varianza non costante degli errori) lungo la diagonale.

In un modello classico con omoschedasticità, Ω sarebbe proporzionale a una matrice identità. Tuttavia, nel caso del metodo Sandwich, Ω può riflettere la variazione e la correlazione effettiva tra gli errori, fornendo così errori standard più realistici.
In pratica, il metodo Sandwich modifica solo il modo in cui vengono calcolati gli errori standard, ma non cambia le stime dei parametri del modello. Questa correzione è particolarmente importante quando le ipotesi di varianza costante e indipendenza degli errori non sono rispettate

Distribuzioni Utilizzate
Di seguito una sintesi delle distribuzioni assunte e della loro applicazione ai metodi sopra descritti:

- **Distribuzione Normale**: Assume che la variabile dipendente segua una distribuzione normale. Anche se i dati di conteggio spesso non seguono una normale, può essere un'approssimazione utile in alcuni contesti. Entrambi i metodi (Wald e Sandwich) sono stati applicati, con il metodo Sandwich che fornisce maggiore robustezza in presenza di deviazioni dalle assunzioni di normalità.
- **Distribuzione Log-normale**: Presuppone che la log-trasformazione della variabile risposta segua una distribuzione normale. Questo approccio è utile quando i dati sono positivamente asimmetrici.
- **Distribuzione di Poisson**: Questa distribuzione è adatta per modellare conteggi. Tuttavia, la distribuzione di Poisson assume che la media sia uguale alla varianza, un'assunzione che spesso viene violata nei dati reali (sovradispersione). Anche in questo caso, l'approccio Sandwich è particolarmente utile per correggere tale violazione.
- **Distribuzione Binomiale negativa**: È una generalizzazione della distribuzione di Poisson che permette di gestire la sovradispersione (quando la varianza supera la media). Questo modello è spesso preferibile per dati di conteggio che mostrano elevata variabilità.

La simulazione utilizzata è la seguente:

1. Uguale allo step 1 descritto in precedenza in questo capitolo
2. Uguale allo step 2 descritto in precedenza in questo capitolo
3. Per ogni coppia di dataset VNS0001-SOC0001, VNS0002-SOC0002 … VNS10000-SOC10000, è stata calcolata la differenza ponderata nella media della durata della risposta massima nei due gruppi di trattamento, insieme ai suoi intervalli di confidenza al 95% calcolati con il metodo parametrico.

   Codice SAS per il metodo di Wald:

```
proc genmod data=outboot_<n>;
class TrtGroup;
model dur36=TrtGroup/dist=<xxx> waldci;
weight olw;
run;
```

Codice SAS per il metodo di Sandwich:

```
proc genmod data=outboot_<n>;
class TrtGroup SubjId;
model dur36=TrtGroup/dist=<xxx>;
repeated subject=SubjId;
weight olw;
run;
```

dove SubjID è l'identificativo del soggetto

4. Se il limite inferiore dell'intervallo di confidenza n-esimo è > 0, allora l'n-esima simulazione è stata considerata un successo. La potenza dell'analisi con V soggetti VNS e S soggetti SoC è stata calcolata dividendo il numero di successi per le 10.000 simulazioni effettuate.

I passaggi 1-4 sono stati ripetuti con numeri crescenti di V e S fino a quando la potenza non è stata ≥ 80%. Il rapporto V:S è stato impostato a 1:2, come descritto nel primo step, in quanto il numero di soggetti disposti all'impianto è solitamente circa la metà di quelli che intendono iniziare/continuare una terapia medica.

Una volta calcolato il sample size con i vari approcci, è stato necessario comprendere come verificare quale approccio fosse corretto. Come soluzione è stata adottata una simulazione per verificare l'inflazione dell'errore di tipo 1. L'errore di tipo 1 dovrebbe essere vicino a 0.025, come stabilito durante il calcolo della numerosità campionaria.
Gli step per calcolare l'errore di tipo 1 sono identici a quelli descritti in questa sezione, con la sola differenza che al posto di campionare dalle popolazioni VNS e SOC, i campionamenti sono stati effettuati da un'unica popolazione frutto dell'unione delle popolazioni VNS e SOC. Campionando quindi da un'unica popolazione, la probabilità di trovare simulazioni significative dovrebbe essere pari all'errore di tipo 1, e se questo non avviene ci si trova di fronte a una deflazione o un'inflazione dell'errore di tipo 1 [27].

Risultati

I risultati ottenuti sono riportati in Figura 2 - Risultati confronto eTabella 6 - Risultati.

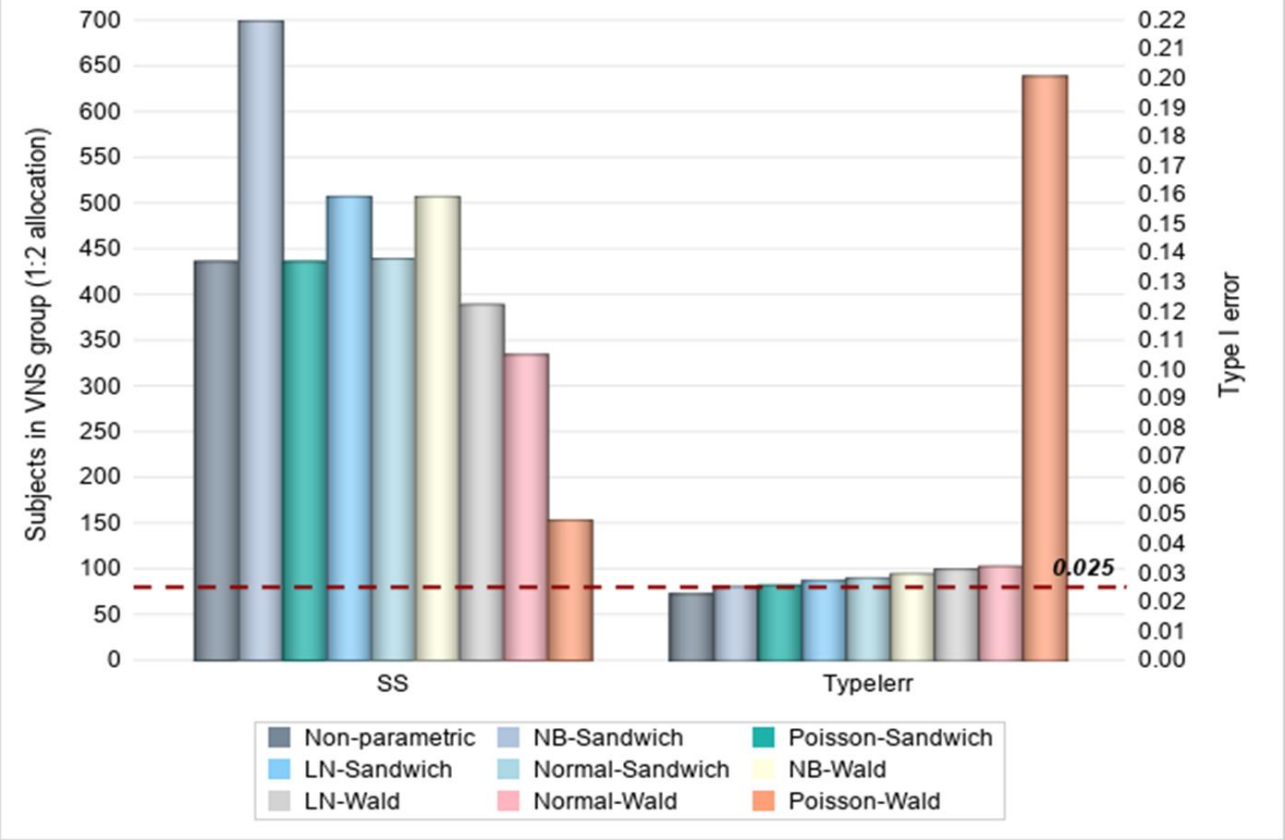

Figura 2 - Risultati confronto

Tabella 6 - Risultati

Distribuzione assunta	Calcolo degli intervalli di confidenza	Sample size VNS	Sample Size SoC	Type I error
Nessuna (approccio non parametrico)	Bootstrapping	437	874	0.0230
Binomiale negativa	Sandwich	700	1400	0.0255
Poisson	Sandwich	437	874	0.0260
Log-normale	Sandwich	508	1016	0.0275
Normale	Sandwich	440	880	0.0285
Binomiale negativa	Wald	508	1016	0.0300
Log-normale	Wald	390	780	0.0315
Normale	Wald	335	670	0.0325
Poisson	Wald	154	308	0.2010

La Figura 2 - Risultati confronto riporta a sinistra il numero di pazienti necessari ad ottenere una potenza dell'80% per ognuno dei metodi considerati, mentre a destra sono riportati i valori dell'errore di tipo 1. Come atteso, il risultato peggiore è stato ottenuto utilizzando la stima di Wald per gli intervalli di confidenza: l'errore di tipo 1 varia dal 3.15% a più del 20.10%, quest'ultimo ottenuto utilizzando la distribuzione di Poisson. L'errore di tipo 1 ottenuto con gli altri metodi si colloca invece

al di sotto del 2.85%. Il bootstrapping non-parametrico permette di ottenere un errore di tipo 1 del 2.30%, mentre i risultati ottenuti utilizzando i Sandwich robust estimator assumendo una distribuzione Poisson e Binomiale Negativa risultano rispettivamente 2.60 e 2.55%. L'assunzione Binomiale Negativa ha richiesto tuttavia un numero di pazienti VNS molto più alto rispetto alla metodologia non parametrica (700 contro 437), per cui i due approcci che hanno funzionato meglio in questa simulazione sono stati l'assunzione della distribuzione Poisson per la variabile risposta unita allo stimatore robusto di Sandwich e il metodo bootstrapping per il calcolo degli intervalli di confidenza, entrambi richiedendo 437 soggetti nel gruppo dei trattati con VNS.
Per la pianificazione dello studio è stato quindi deciso di procedere utilizzando per l'endpoint primario un'inferenza basata sul bootstrapping, in quanto l'assunzione di Poisson, seppur dando risultati apprezzabili, è ben lontana dalla realtà mostrata dalla Figura 1 - Distribuzione della durata massima della risposta al trattamento.

Discussione

Implicazioni dei Risultati

I risultati di questa ricerca evidenziano l'importanza di selezionare tecniche appropriate per fare inferenza quando si confrontano campioni con dati pesati. Il bootstrapping non-parametrico emerge come la tecnica più robusta e affidabile in caso siano a disposizione gli IPD, ma le tecniche basate su sandwich robust estimator offrono una buona alternativa, specialmente in contesti con dati distribuiti secondo Poisson o Binomiale Negativa, sebbene possano richiedere campioni più grandi. Questi ultimi risultano inoltre più rapidi nell'esecuzione rispetto al bootstrapping, aspetto non secondario quando si lavora con le sperimentazioni cliniche.

Limitazioni dello Studio

La limitazione principale di questa ricerca è rappresentata dalla specificità del caso di studio utilizzato. Sebbene le tecniche sviluppate siano applicabili in una vasta gamma di contesti clinici, i risultati ottenuti sono specifici per lo scenario del trattamento dell'epilessia con VNS e non sono purtroppo generalizzabili ad altri contesti.
Inoltre, le simulazioni condotte si basano su assunzioni specifiche sui dati simulati. L'accuratezza delle inferenze potrebbe variare in presenza di dati reali con complessità aggiuntive non catturate nelle simulazioni.
La presente simulazione non permette inoltre di valutare in che misura la numerosità campionaria ottenuta con la distribuzione binomiale negativa sia sensibile al grado di aderenza dei dati a tale distribuzione.
A differenza di quanto avviene in molte simulazioni, dove l'effetto del trattamento viene fissato a priori per valutare le prestazioni dei metodi statistici, nel presente studio si è partiti da una differenza media osservata pari a 3.15 tra i due gruppi, che rappresenta però un confronto osservazionale in assenza di randomizzazione. Pertanto, l'effettiva grandezza dell'effetto tra popolazioni realmente comparabili (sebbene stimata in Tabella 4) non è nota, e ciò costituisce una limitazione dello studio.

Tale scelta non è stata arbitraria, ma vincolata dai dati disponibili, forniti dai clinici, che rappresentavano l'unico punto di partenza realistico per lo sviluppo della simulazione.

Conclusioni

Questa tesi ha esplorato tecniche avanzate per fare inferenza utilizzando dati IPD. Le simulazioni condotte hanno dimostrato che un approccio personalizzato in base al contesto è necessario, e si consiglia di eseguire simulazioni similari ogni qual volta si presenti una richiesta di pianificare uno studio in un contesto non randomizzato.
Purtroppo le simulazioni utilizzate in questo progetto sono risultate time-consuming, e non sarebbe fattibile applicarle ad una richiesta lavorativa per la quale si abbia una tempistica limitata nel tempo. Ulteriori sviluppi potrebbero essere quindi l'ottimizzazione dei programmi utilizzati in modo da ridurre le tempistiche, oltre che l'applicazione delle stesse simulazioni a popolazioni diverse per verificare la robustezza dei risultati ottenuti in altri contesti.

Riferimenti

1. Howland RH. Vagus Nerve Stimulation. *Curr Behav Neurosci Rep*. 2014;1(2):64-73.
2. Kwan P, Arzimanoglou A, Berg AT, et al. "Definition of drug resistant epilepsy: Consensus proposal by the ad hoc Task Force of the ILAE Commission on Therapeutic Strategies." *Epilepsia*. 2010;51(6):1069–1077.
3. Dalic L, Cook MJ. "Managing drug-resistant epilepsy: challenges and solutions." *Neuropsychiatric Disease and Treatment.* 2016;12:2605–2616.
4. Salles, Gilles, et al. "Tafasitamab plus lenalidomide in relapsed or refractory diffuse large B-cell lymphoma (L-MIND): a multicentre, prospective, single-arm, phase 2 study." *The Lancet Oncology* 21.7 (2020): 978-988.
5. Zinzani PL, Rodgers T, Marino D, et al. RE-MIND: Comparing Tafasitamab + Lenalidomide (L-MIND) with a Real-world Lenalidomide Monotherapy Cohort in Relapsed or Refractory Diffuse Large B-cell Lymphoma. *Clin Cancer Res*. 2021;27(22):6124-6134
6. Austin PC. "An introduction to propensity score methods for reducing the effects of confounding in observational studies." *Multivariate Behavioral Research*. 2011;46(3): 399-424.
7. Serrano P, Yuen HW, Akdemir J, et al. Real-world data in drug development strategies for orphan drugs: Tafasitamab in B-cell lymphoma, a case study for an approval based on a single-arm combination trial. *Drug Discov Today*. 2022;27(6):1706-1715
8. Austin PC, Mamdani MM. A comparison of propensity score methods: a case-study estimating the effectiveness of post-AMI statin use. *Stat Med.* 2006;25:2084-2106.
9. Austin P.C. Propensity-score matching in the cardiovascular surgery literature from 2004 to 2006: A systematic review and suggestions for improvement. *Journal of Thoracic and Cardiovascular Surgery.* 2007a;134:1128–1135.
10. Austin P.C. The performance of different propensity score methods for estimating marginal odds ratios. *Statistics in Medicine.* 2007b;26:3078–3094.
11. Austin P.C. The performance of different propensity score methods for estimating relative risks. *Journal of Clinical Epidemiology.* 2008a;61:537–545.
12. Austin P.C. A critical appraisal of propensity score matching in the medical literature from 1996 to 2003. *Statistics in Medicine.* 2008b;27:2037–2049.
13. Austin P.C. A report card on propensity-score matching in the cardiology literature from 2004 to 2006: Results of a systematic review. *Circulation: Cardiovascular Quality and Outcomes.* 2008c;1:62–67.
14. Schafer JL, Kang J. Average causal effects from nonrandomized studies: a practical guide and simulated example. *Psychological Methods*. 2008;13:279–313.
15. Austin PC. Type I error rates, coverage of confidence intervals, and variance estimation in propensity-score matched analyses. *The International Journal of Biostatistics*. 2009;5.
16. Austin, Peter C. “Comparing paired vs non-paired statistical methods of analyses when making inferences about absolute risk reductions in propensity-score matched samples.” *Statistics in medicine* vol. 30,11 (2011): 1292-301.
17. Austin P.C. Goodness-of-fit diagnostics for the propensity score model when estimating treatment effects using covariate adjustment with the propensity score. *Pharmacoepidemiology and Drug Safety.* 2008e;17:1202–1217.
18. Austin P.C. The relative ability of different propensity-score methods to balance measured covariates between treated and untreated subjects in observational studies. *Medical Decision Making.* 2009b;29:661–677.

19. Austin P.C. Using the standardized difference to compare the prevalence of a binary variable between two groups in observational research. Communications in Statistics—*Simulation and Computation.* 2009d;38:1228–1234.
20. Austin P.C. The performance of different propensity score methods for estimating difference in proportions (risk differences or absolute risk reductions) in observational studies. *Statistics in Medicine.* 2010;29:2137–2148.
21. Austin PC. An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies. *Multivariate Behav Res*. 2011;46(3):399-424
22. Austin PC. Variance estimation when using inverse probability of treatment weighting (IPTW) with survival analysis. *Stat Med*. 2016;35(30):5642-5655.
23. Gayat E, Resche-Rigon M, Mary JY, Porcher R. Propensity score applied to survival data analysis through proportional hazards models: a Monte Carlo study. *Pharmaceutical Statistics* 2012; 11(3):222–229. 8.
24. Austin PC. The performance of different propensity score methods for estimating marginal hazard ratios. *Stastisics in Medicine* 2013; 32(16):2837–2849.
25. Abadie A, Imbens GW. Notes and comments on the failure of the bootstrap for matching estimators. Econometrica 2008; 76(6):1537–1557.
26. Austin PC, Small DS. The use of bootstrapping when using propensity-score matching without replacement: a simulation study. Statisics in Medicine 2014; 33(24):4306–4319.
27. Lunceford JK, Davidian M. Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study. Statistics in Medicine 2004; 23(19):2937–2960.
28. Joffe MM, Ten Have TR, Feldman HI, Kimmel SE. Model selection, confounder control, and marginal structural models: review and new applications. The American Statistician 2004; 58:272–279.
29. Xu S, Ross C, Raebel MA, Shetterly S, Blanchette C, Smith D. Use of stabilized inverse propensity scores as weights to directly estimate relative risk and its confidence intervals. Value in Health 2010; 13(2):273–277.
30. Williamson EJ, Forbes A, White IR. Variance reduction in randomised trials by inverse probability weighting using the propensity score. Statistics in Medicine 2014; 33(5):721–737.
31. Hernan MA, Brumback B, Robins JM. Marginal structural models to estimate the causal effect of zidovudine on the survival of HIV-positive men. Epidemiology 2000; 11(5):561–570
32. Li, F., Morgan, K. L., & Zaslavsky, A. M. (2018). Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, 113(521), 390-400
33. Rosenbaum, P. R. (2012). Optimal Matching of an Optimally Chosen Subset in Observational Studies. *Journal of Computational and Graphical Statistics*, 21(1), 57–71
34. Busso, M., DiNardo, J.,and McCrary, J.(2014), New Evidence on the Finite Sample Properties of Propensity Score Reweighting and Matching Estimators. *The Review of Economics and Statistics* ,96, 885-897. [391]
35. Agresti, A. (2013). *Categorical Data Analysis (3rd ed.)*. John Wiley & Sons
36. Kauermann, G., & Carroll, R. J. (2000). The sandwich variance estimator: Efficiency properties and coverage probability of confidence intervals.

Appendice A: codice SAS per la creazione delle popolazioni

```
libname pop 'S:\Rep_Client_Project_current\Simpop';

/*ASM population high discontinuation rate, 7.7% of new responders*/
%macro ASM_highdisc7s;
data ASM_highdisc7s;
call streaminit(1);
     do x = 1 to 1000000;

     nsucc1=50;
     nsucc2=50;
     nsucc3=50;
     nsucc6=40;
     nsucc12=35;
     nsucc24=35;
     nsucc36=35;
     nsucc4=nsucc3+((nsucc6-nsucc3)/3);
     nsucc5=nsucc4+((nsucc6-nsucc3)/3);
     nsucc7=nsucc6+((nsucc12-nsucc6)/6);
     nsucc8=nsucc7+((nsucc12-nsucc6)/6);
     nsucc9=nsucc8+((nsucc12-nsucc6)/6);
     nsucc10=nsucc9+((nsucc12-nsucc6)/6);
     nsucc11=nsucc10+((nsucc12-nsucc6)/6);
     nsucc13=nsucc12+((nsucc24-nsucc12)/12);
     nsucc14=nsucc13+((nsucc24-nsucc12)/12);
     nsucc15=nsucc14+((nsucc24-nsucc12)/12);
     nsucc16=nsucc15+((nsucc24-nsucc12)/12);
     nsucc17=nsucc16+((nsucc24-nsucc12)/12);
     nsucc18=nsucc17+((nsucc24-nsucc12)/12);
```

```
nsucc19=nsucc18+((nsucc24-nsucc12)/12);
nsucc20=nsucc19+((nsucc24-nsucc12)/12);
nsucc21=nsucc20+((nsucc24-nsucc12)/12);
nsucc22=nsucc21+((nsucc24-nsucc12)/12);
nsucc23=nsucc22+((nsucc24-nsucc12)/12);
nsucc25=nsucc24+((nsucc36-nsucc24)/12);
nsucc26=nsucc25+((nsucc36-nsucc24)/12);
nsucc27=nsucc26+((nsucc36-nsucc24)/12);
nsucc28=nsucc27+((nsucc36-nsucc24)/12);
nsucc29=nsucc28+((nsucc36-nsucc24)/12);
nsucc30=nsucc29+((nsucc36-nsucc24)/12);
nsucc31=nsucc30+((nsucc36-nsucc24)/12);
nsucc32=nsucc31+((nsucc36-nsucc24)/12);
nsucc33=nsucc32+((nsucc36-nsucc24)/12);
nsucc34=nsucc33+((nsucc36-nsucc24)/12);
nsucc35=nsucc34+((nsucc36-nsucc24)/12);

ndisc1=5;
ndisc2=10;
ndisc3=15;
ndisc6=25;
ndisc12=40;
ndisc24=50;
ndisc36=65;
ndisc4=ndisc3+((ndisc6-ndisc3)/3);
ndisc5=ndisc4+((ndisc6-ndisc3)/3);
ndisc7=ndisc6+((ndisc12-ndisc6)/6);
ndisc8=ndisc7+((ndisc12-ndisc6)/6);
ndisc9=ndisc8+((ndisc12-ndisc6)/6);
ndisc10=ndisc9+((ndisc12-ndisc6)/6);
ndisc11=ndisc10+((ndisc12-ndisc6)/6);
```

```
 ndisc13=ndisc12+((ndisc24-ndisc12)/12);
 ndisc14=ndisc13+((ndisc24-ndisc12)/12);
 ndisc15=ndisc14+((ndisc24-ndisc12)/12);
 ndisc16=ndisc15+((ndisc24-ndisc12)/12);
 ndisc17=ndisc16+((ndisc24-ndisc12)/12);
 ndisc18=ndisc17+((ndisc24-ndisc12)/12);
 ndisc19=ndisc18+((ndisc24-ndisc12)/12);
 ndisc20=ndisc19+((ndisc24-ndisc12)/12);
 ndisc21=ndisc20+((ndisc24-ndisc12)/12);
 ndisc22=ndisc21+((ndisc24-ndisc12)/12);
 ndisc23=ndisc22+((ndisc24-ndisc12)/12);
 ndisc25=ndisc24+((ndisc36-ndisc24)/12);
 ndisc26=ndisc25+((ndisc36-ndisc24)/12);
 ndisc27=ndisc26+((ndisc36-ndisc24)/12);
 ndisc28=ndisc27+((ndisc36-ndisc24)/12);
 ndisc29=ndisc28+((ndisc36-ndisc24)/12);
 ndisc30=ndisc29+((ndisc36-ndisc24)/12);
 ndisc31=ndisc30+((ndisc36-ndisc24)/12);
 ndisc32=ndisc31+((ndisc36-ndisc24)/12);
 ndisc33=ndisc32+((ndisc36-ndisc24)/12);
 ndisc34=ndisc33+((ndisc36-ndisc24)/12);
 ndisc35=ndisc34+((ndisc36-ndisc24)/12);

m1 = rand("bernoulli", nsucc1/100);
 m1 = m1+1;
 if m1=1 then m1=rand("bernoulli", 1/9);
    if m1=1 then m1=.;
    if m1=2 then m1=1;

 %do i=1 %to 35;
 %let j=%eval(&i.+1);
```

```
     if m&i.=. then m&j.=.;
        else if m&i.=0 then m&j.= rand("bernoulli", 0.077*(1-((ndisc&j.-ndisc&i.)/(nsucc&i.-
nsucc&j.+(100-nsucc&i.-ndisc&i.)))));
              else m&j. = rand("bernoulli", ((nsucc&j.-0.077*(100-nsucc&i.-ndisc&i.)*(1-
((ndisc&j.-ndisc&i.)/(nsucc&i.-nsucc&j.+(100-nsucc&i.-ndisc&i.)))))/nsucc&i.));
     m&j. = m&j.+1;
     if m&j.=1 then m&j.=rand("bernoulli", (ndisc&j.-ndisc&i.)/(nsucc&i.-nsucc&j.+(100-nsucc&i.-
ndisc&i.)));
     if m&j.=1 then m&j.=.;
     if m&j.=2 then m&j.=1;
     %end;
     group='B';
     output;
     end;
run;
%mend;

%ASM_highdisc7s;

data ASM_highdisc7s;
set ASM_highdisc7s;
total12=sum(of m1-m12);
total24=sum(of m1-m24);
total36=sum(of m1-m36);
if total12=. then total12=0;
if total24=. then total24=0;
if total36=. then total36=0;
run;

data temp12;
set ASM_highdisc7s;
```

```
if total12=0 then output;
array m [*] m1-m12;
     if m[1] = 1 then do; start=1; output; end;
     do i=2 to dim(m);
          if m[i] = 1 and m[i-1] = 0 then do; start=i; output; end;
     end;
     if m[12] = 1 then do; end=12; output; end;
     do i=1 to dim(m)-1;
          if m[dim(m)-i] = 1 and (m[dim(m)-i+1] = 0 or m[dim(m)-i+1] = .) then do; end=dim(m)-i;
output; end;
     end;
keep x start end;
run;

data start12;
set temp12;
if end=.;
keep x start;
data end12;
set temp12;
if start=. or end^=.;
keep x end;
run;

proc sort data=end12;
by x end;
run;

data dur12;
set start12;
set end12;
```

```
dur12=end-start+1;
keep x dur12;
run;

proc sort data=dur12;
by x descending dur12;
run;

proc sort data=dur12 nodupkey;
by x;
run;


data temp24;
set ASM_highdisc7s;
if total24=0 then output;
array m [*] m1-m24;
     if m[1] = 1 then do; start=1; output; end;
     do i=2 to dim(m);
          if m[i] = 1 and m[i-1] = 0 then do; start=i; output; end;
     end;
     if m[24] = 1 then do; end=24; output; end;
     do i=1 to dim(m)-1;
          if m[dim(m)-i] = 1 and (m[dim(m)-i+1] = 0 or m[dim(m)-i+1] = .) then do; end=dim(m)-i;
output; end;
     end;
keep x start end;
run;

data start24;
set temp24;
```

```
if end=.;
keep x start;
data end24;
set temp24;
if start=. or end^=.;
keep x end;
run;

proc sort data=end24;
by x end;
run;

data dur24;
set start24;
set end24;
dur24=end-start+1;
keep x dur24;
run;

proc sort data=dur24;
by x descending dur24;
run;

proc sort data=dur24 nodupkey;
by x;
run;


data temp36;
set ASM_highdisc7s;
if total36=0 then output;
```

```
array m [*] m1-m36;
     if m[1] = 1 then do; start=1; output; end;
     do i=2 to dim(m);
          if m[i] = 1 and m[i-1] = 0 then do; start=i; output; end;
     end;
     if m[36] = 1 then do; end=36; output; end;
     do i=1 to dim(m)-1;
          if m[dim(m)-i] = 1 and (m[dim(m)-i+1] = 0 or m[dim(m)-i+1] = .) then do; end=dim(m)-i;
output; end;
     end;
keep x start end;
run;

data start36;
set temp36;
if end=.;
keep x start;
data end36;
set temp36;
if start=. or end^=.;
keep x end;
run;

proc sort data=end36;
by x end;
run;

data dur36;
set start36;
set end36;
dur36=end-start+1;
```

```
keep x dur36;
run;

proc sort data=dur36;
by x descending dur36;
run;

proc sort data=dur36 nodupkey;
by x;
run;

data pop.ASM_highdisc7s;
set ASM_highdisc7s;
set dur12;
set dur24;
set dur36;
if total12=0 then dur12=0;
if total24=0 then dur24=0;
if total36=0 then dur36=0;
run;

ods pdf file="S:\Rep_Client_Project_current\Simpop\VNS_pop_desc.pdf";
title "Frequency of missings, non-responders and responders in VNS population";
proc freq data=pop.ASM_highdisc7s;
table m1-m36/missing;
run;

title "Distribution of duration of response in VNS population";
proc univariate data=pop.ASM_highdisc7s;
var dur12 dur24 dur36 total12 total24 total36;
run;
```

```
ods pdf close;

proc univariate data=pop.ASM_highdisc7s;
var dur12 dur24 dur36 total12 total24 total36;
run;


proc datasets lib=work nolist memtype=data kill;
run;
quit;



/*VNS population medium discontinuation rate based on CORE*/
data VNS_mediumCORE;
call streaminit(1);
     do x = 1 to 1000000;
        m3cont=0.98;
        m6cont=0.96;
        m12cont=0.94;
        m24cont=0.88;
        m36cont=0.81;
        m3cadj=m3cont**(1/3);
        m6cadj=(m6cont/m3cont)**(1/3);
        m12cadj=(m12cont/m6cont)**(1/6);
        m24cadj=(m24cont/m12cont)**(1/12);
        m36cadj=(m36cont/m24cont)**(1/12);

        m1 = rand("bernoulli", m3cadj);
```

```
if m1=0 then m1=.;
if m1=1 then m1=rand("bernoulli", 0.24103);

if m1=. then m2=.;
else m2 = rand("bernoulli", m3cadj);
if m2=0 then m2=.;
if m2=1 then do;
       if m1 = 1 then m2=rand("bernoulli", 0.74286);
       if m1 = 0 then m2=rand("bernoulli", 0.12745);
end;

if m2=. then m3=.;
else m3 = rand("bernoulli", m3cadj);
if m3=0 then m3=.;
if m3=1 then do;
       if m1 = 1 and m2 = 1 then m3=rand("bernoulli", 0.69231);
       if m1 = 1 and m2 = 0 then m3=rand("bernoulli", 0.33333);
       if m1 = 0 and m2 = 1 then m3=rand("bernoulli", 0.33333);
       if m1 = 0 and m2 = 0 then m3=rand("bernoulli", 0.22472);
end;

if m3=. then m4=.;
else m4 = rand("bernoulli", m6cadj);
if m4=0 then m4=.;
if m4=1 then do;
       if m3 = 1 then m4=rand("bernoulli", 0.92041);
       if m3 = 0 then m4=rand("bernoulli", 0.07493);
end;

if m4=. then m5=.;
else m5 = rand("bernoulli", m6cadj);
```

```
if m5=0 then m5=.;
if m5=1 then do;
       if m4 = 1 then m5=rand("bernoulli", 0.92041);
       if m4 = 0 then m5=rand("bernoulli", 0.07493);
end;

if m5=. then m6=.;
else m6 = rand("bernoulli", m6cadj);
if m6=0 then m6=.;
if m6=1 then do;
       if m5 = 1 then m6=rand("bernoulli", 0.92041);
       if m5 = 0 then m6=rand("bernoulli", 0.07493);
end;

if m6=. then m7=.;
else m7 = rand("bernoulli", m12cadj);
if m7=0 then m7=.;
if m7=1 then do;
       if m5 = 1 and m6 = 1 then m7=rand("bernoulli", 0.98310);
       if m5 = 1 and m6 = 0 then m7=rand("bernoulli", 0.08744);
       if m5 = 0 and m6 = 1 then m7=rand("bernoulli", 0.95899);
       if m5 = 0 and m6 = 0 then m7=rand("bernoulli", 0.03853);
end;

if m7=. then m8=.;
else m8 = rand("bernoulli", m12cadj);
if m8=0 then m8=.;
if m8=1 then do;
       if m6 = 1 and m7 = 1 then m8=rand("bernoulli", 0.98310);
       if m6 = 1 and m7 = 0 then m8=rand("bernoulli", 0.08744);
       if m6 = 0 and m7 = 1 then m8=rand("bernoulli", 0.95899);
```

```
        if m6 = 0 and m7 = 0 then m8=rand("bernoulli", 0.03853);
end;

if m8=. then m9=.;
else m9 = rand("bernoulli", m12cadj);
if m9=0 then m9=.;
if m9=1 then do;
        if m7 = 1 and m8 = 1 then m9=rand("bernoulli", 0.98310);
        if m7 = 1 and m8 = 0 then m9=rand("bernoulli", 0.08744);
        if m7 = 0 and m8 = 1 then m9=rand("bernoulli", 0.95899);
        if m7 = 0 and m8 = 0 then m9=rand("bernoulli", 0.03853);
end;

if m9=. then m10=.;
else m10 = rand("bernoulli", m12cadj);
if m10=0 then m10=.;
if m10=1 then do;
        if m8 = 1 and m9 = 1 then m10=rand("bernoulli", 0.98310);
        if m8 = 1 and m9 = 0 then m10=rand("bernoulli", 0.08744);
        if m8 = 0 and m9 = 1 then m10=rand("bernoulli", 0.95899);
        if m8 = 0 and m9 = 0 then m10=rand("bernoulli", 0.03853);
end;

if m10=. then m11=.;
else m11 = rand("bernoulli", m12cadj);
if m11=0 then m11=.;
if m11=1 then do;
        if m9 = 1 and m10 = 1 then m11=rand("bernoulli", 0.98310);
        if m9 = 1 and m10 = 0 then m11=rand("bernoulli", 0.08744);
        if m9 = 0 and m10 = 1 then m11=rand("bernoulli", 0.95899);
        if m9 = 0 and m10 = 0 then m11=rand("bernoulli", 0.03853);
```

```
end;

if m11=. then m12=.;
else m12 = rand("bernoulli", m12cadj);
if m12=0 then m12=.;
if m12=1 then do;
       if m10 = 1 and m11 = 1 then m12=rand("bernoulli", 0.98310);
       if m10 = 1 and m11 = 0 then m12=rand("bernoulli", 0.08744);
       if m10 = 0 and m11 = 1 then m12=rand("bernoulli", 0.95899);
       if m10 = 0 and m11 = 0 then m12=rand("bernoulli", 0.03853);
end;

if m12=. then m13=.;
else m13 = rand("bernoulli", m24cadj);
if m13=0 then m13=.;
if m13=1 then do;
       if m11 = 1 and m12 = 1 then m13=rand("bernoulli", 0.97132);
       if m11 = 1 and m12 = 0 then m13=rand("bernoulli", 0.05523);
       if m11 = 0 and m12 = 1 then m13=rand("bernoulli", 0.89875);
       if m11 = 0 and m12 = 0 then m13=rand("bernoulli", 0.03703);
end;

if m13=. then m14=.;
else m14 = rand("bernoulli", m24cadj);
if m14=0 then m14=.;
if m14=1 then do;
       if m12 = 1 and m13 = 1 then m14=rand("bernoulli", 0.97132);
       if m12 = 1 and m13 = 0 then m14=rand("bernoulli", 0.05523);
       if m12 = 0 and m13 = 1 then m14=rand("bernoulli", 0.89875);
       if m12 = 0 and m13 = 0 then m14=rand("bernoulli", 0.03703);
end;
```

```
if m14=. then m15=.;
else m15 = rand("bernoulli", m24cadj);
if m15=0 then m15=.;
if m15=1 then do;
       if m13 = 1 and m14 = 1 then m15=rand("bernoulli", 0.97132);
       if m13 = 1 and m14 = 0 then m15=rand("bernoulli", 0.05523);
       if m13 = 0 and m14 = 1 then m15=rand("bernoulli", 0.89875);
       if m13 = 0 and m14 = 0 then m15=rand("bernoulli", 0.03703);
end;

if m15=. then m16=.;
else m16 = rand("bernoulli", m24cadj);
if m16=0 then m16=.;
if m16=1 then do;
       if m14 = 1 and m15 = 1 then m16=rand("bernoulli", 0.97132);
       if m14 = 1 and m15 = 0 then m16=rand("bernoulli", 0.05523);
       if m14 = 0 and m15 = 1 then m16=rand("bernoulli", 0.89875);
       if m14 = 0 and m15 = 0 then m16=rand("bernoulli", 0.03703);
end;

if m16=. then m17=.;
else m17 = rand("bernoulli", m24cadj);
if m17=0 then m17=.;
if m17=1 then do;
       if m15 = 1 and m16 = 1 then m17=rand("bernoulli", 0.97132);
       if m15 = 1 and m16 = 0 then m17=rand("bernoulli", 0.05523);
       if m15 = 0 and m16 = 1 then m17=rand("bernoulli", 0.89875);
       if m15 = 0 and m16 = 0 then m17=rand("bernoulli", 0.03703);
end;
```

```
if m17=. then m18=.;
else m18 = rand("bernoulli", m24cadj);
if m18=0 then m18=.;
if m18=1 then do;
       if m16 = 1 and m17 = 1 then m18=rand("bernoulli", 0.97132);
       if m16 = 1 and m17 = 0 then m18=rand("bernoulli", 0.05523);
       if m16 = 0 and m17 = 1 then m18=rand("bernoulli", 0.89875);
       if m16 = 0 and m17 = 0 then m18=rand("bernoulli", 0.03703);
end;

if m18=. then m19=.;
else m19 = rand("bernoulli", m24cadj);
if m19=0 then m19=.;
if m19=1 then do;
       if m17 = 0 and m18 = 1 then m19=rand("bernoulli", 0.92074);
       if m17 = 0 and m18 = 0 then m19=rand("bernoulli", 0.04002);
       if m16 = 1 and m17 = 1 and m18 = 1 then m19=rand("bernoulli", 0.98205);
       if m16 = 1 and m17 = 1 and m18 = 0 then m19=rand("bernoulli", 0.10660);
       if m16 = 0 and m17 = 1 and m18 = 1 then m19=rand("bernoulli", 0.96502);
       if m16 = 0 and m17 = 1 and m18 = 0 then m19=rand("bernoulli", 0.05832);
       if m16 = 0 and m17 = 0 and m18 = 1 then m19=rand("bernoulli", 0.93009);
       if m16 = 0 and m17 = 0 and m18 = 0 then m19=rand("bernoulli", 0.03640);
end;

if m19=. then m20=.;
else m20 = rand("bernoulli", m24cadj);
if m20=0 then m20=.;
if m20=1 then do;
       if m18 = 0 and m19 = 1 then m20=rand("bernoulli", 0.92074);
       if m18 = 0 and m19 = 0 then m20=rand("bernoulli", 0.04002);
       if m17 = 1 and m18 = 1 and m19 = 1 then m20=rand("bernoulli", 0.98205);
```

```
        if m17 = 1 and m18 = 1 and m19 = 0 then m20=rand("bernoulli", 0.10660);
        if m17 = 0 and m18 = 1 and m19 = 1 then m20=rand("bernoulli", 0.96502);
        if m17 = 0 and m18 = 1 and m19 = 0 then m20=rand("bernoulli", 0.05832);
        if m17 = 0 and m18 = 0 and m19 = 1 then m20=rand("bernoulli", 0.93009);
        if m17 = 0 and m18 = 0 and m19 = 0 then m20=rand("bernoulli", 0.03640);
end;

if m20=. then m21=.;
else m21 = rand("bernoulli", m24cadj);
if m21=0 then m21=.;
if m21=1 then do;
        if m19 = 0 and m20 = 1 then m21=rand("bernoulli", 0.92074);
        if m19 = 0 and m20 = 0 then m21=rand("bernoulli", 0.04002);
        if m18 = 1 and m19 = 1 and m20 = 1 then m21=rand("bernoulli", 0.98205);
        if m18 = 1 and m19 = 1 and m20 = 0 then m21=rand("bernoulli", 0.10660);
        if m18 = 0 and m19 = 1 and m20 = 1 then m21=rand("bernoulli", 0.96502);
        if m18 = 0 and m19 = 1 and m20 = 0 then m21=rand("bernoulli", 0.05832);
        if m18 = 0 and m19 = 0 and m20 = 1 then m21=rand("bernoulli", 0.93009);
        if m18 = 0 and m19 = 0 and m20 = 0 then m21=rand("bernoulli", 0.03640);
end;

if m21=. then m22=.;
else m22 = rand("bernoulli", m24cadj);
if m22=0 then m22=.;
if m22=1 then do;
        if m20 = 0 and m21 = 1 then m22=rand("bernoulli", 0.92074);
        if m20 = 0 and m21 = 0 then m22=rand("bernoulli", 0.04002);
        if m19 = 1 and m20 = 1 and m21 = 1 then m22=rand("bernoulli", 0.98205);
        if m19 = 1 and m20 = 1 and m21 = 0 then m22=rand("bernoulli", 0.10660);
        if m19 = 0 and m20 = 1 and m21 = 1 then m22=rand("bernoulli", 0.96502);
        if m19 = 0 and m20 = 1 and m21 = 0 then m22=rand("bernoulli", 0.05832);
```

```
        if m19 = 0 and m20 = 0 and m21 = 1 then m22=rand("bernoulli", 0.93009);
        if m19 = 0 and m20 = 0 and m21 = 0 then m22=rand("bernoulli", 0.03640);
end;

if m22=. then m23=.;
else m23 = rand("bernoulli", m24cadj);
if m23=0 then m23=.;
if m23=1 then do;
        if m21 = 0 and m22 = 1 then m23=rand("bernoulli", 0.92074);
        if m21 = 0 and m22 = 0 then m23=rand("bernoulli", 0.04002);
        if m20 = 1 and m21 = 1 and m22 = 1 then m23=rand("bernoulli", 0.98205);
        if m20 = 1 and m21 = 1 and m22 = 0 then m23=rand("bernoulli", 0.10660);
        if m20 = 0 and m21 = 1 and m22 = 1 then m23=rand("bernoulli", 0.96502);
        if m20 = 0 and m21 = 1 and m22 = 0 then m23=rand("bernoulli", 0.05832);
        if m20 = 0 and m21 = 0 and m22 = 1 then m23=rand("bernoulli", 0.93009);
        if m20 = 0 and m21 = 0 and m22 = 0 then m23=rand("bernoulli", 0.03640);
end;

if m23=. then m24=.;
else m24 = rand("bernoulli", m24cadj);
if m24=0 then m24=.;
if m24=1 then do;
        if m22 = 0 and m23 = 1 then m24=rand("bernoulli", 0.92074);
        if m22 = 0 and m23 = 0 then m24=rand("bernoulli", 0.04002);
        if m21 = 1 and m22 = 1 and m23 = 1 then m24=rand("bernoulli", 0.98205);
        if m21 = 1 and m22 = 1 and m23 = 0 then m24=rand("bernoulli", 0.10660);
        if m21 = 0 and m22 = 1 and m23 = 1 then m24=rand("bernoulli", 0.96502);
        if m21 = 0 and m22 = 1 and m23 = 0 then m24=rand("bernoulli", 0.05832);
        if m21 = 0 and m22 = 0 and m23 = 1 then m24=rand("bernoulli", 0.93009);
        if m21 = 0 and m22 = 0 and m23 = 0 then m24=rand("bernoulli", 0.03640);
end;
```

```
if m24=. then m25=.;
else m25 = rand("bernoulli", m36cadj);
if m25=0 then m25=.;
if m25=1 then do;
       if m23 = 0 and m24 = 1 then m25=rand("bernoulli", 0.97044);
       if m23 = 0 and m24 = 0 then m25=rand("bernoulli", 0.01757);
       if m22 = 1 and m23 = 1 and m24 = 1 then m25=rand("bernoulli", 0.99215);
       if m22 = 1 and m23 = 1 and m24 = 0 then m25=rand("bernoulli", 0.05200);
       if m22 = 0 and m23 = 1 and m24 = 1 then m25=rand("bernoulli", 0.98699);
       if m22 = 0 and m23 = 1 and m24 = 0 then m25=rand("bernoulli", 0.04022);
       if m22 = 0 and m23 = 0 and m24 = 1 then m25=rand("bernoulli", 0.96992);
       if m22 = 0 and m23 = 0 and m24 = 0 then m25=rand("bernoulli", 0.01364);
end;

if m25=. then m26=.;
else m26 = rand("bernoulli", m36cadj);
if m26=0 then m26=.;
if m26=1 then do;
       if m24 = 0 and m25 = 1 then m26=rand("bernoulli", 0.97044);
       if m24 = 0 and m25 = 0 then m26=rand("bernoulli", 0.01757);
       if m23 = 1 and m24 = 1 and m25 = 1 then m26=rand("bernoulli", 0.99215);
       if m23 = 1 and m24 = 1 and m25 = 0 then m26=rand("bernoulli", 0.05200);
       if m23 = 0 and m24 = 1 and m25 = 1 then m26=rand("bernoulli", 0.98699);
       if m23 = 0 and m24 = 1 and m25 = 0 then m26=rand("bernoulli", 0.04022);
       if m23 = 0 and m24 = 0 and m25 = 1 then m26=rand("bernoulli", 0.96992);
       if m23 = 0 and m24 = 0 and m25 = 0 then m26=rand("bernoulli", 0.01364);
end;

if m26=. then m27=.;
else m27 = rand("bernoulli", m36cadj);
```

```
if m27=0 then m27=.;
if m27=1 then do;
       if m25 = 0 and m26 = 1 then m27=rand("bernoulli", 0.97044);
       if m25 = 0 and m26 = 0 then m27=rand("bernoulli", 0.01757);
       if m24 = 1 and m25 = 1 and m26 = 1 then m27=rand("bernoulli", 0.99215);
       if m24 = 1 and m25 = 1 and m26 = 0 then m27=rand("bernoulli", 0.05200);
       if m24 = 0 and m25 = 1 and m26 = 1 then m27=rand("bernoulli", 0.98699);
       if m24 = 0 and m25 = 1 and m26 = 0 then m27=rand("bernoulli", 0.04022);
       if m24 = 0 and m25 = 0 and m26 = 1 then m27=rand("bernoulli", 0.96992);
       if m24 = 0 and m25 = 0 and m26 = 0 then m27=rand("bernoulli", 0.01364);
end;

if m27=. then m28=.;
else m28 = rand("bernoulli", m36cadj);
if m28=0 then m28=.;
if m28=1 then do;
       if m26 = 0 and m27 = 1 then m28=rand("bernoulli", 0.97044);
       if m26 = 0 and m27 = 0 then m28=rand("bernoulli", 0.01757);
       if m25 = 1 and m26 = 1 and m27 = 1 then m28=rand("bernoulli", 0.99215);
       if m25 = 1 and m26 = 1 and m27 = 0 then m28=rand("bernoulli", 0.05200);
       if m25 = 0 and m26 = 1 and m27 = 1 then m28=rand("bernoulli", 0.98699);
       if m25 = 0 and m26 = 1 and m27 = 0 then m28=rand("bernoulli", 0.04022);
       if m25 = 0 and m26 = 0 and m27 = 1 then m28=rand("bernoulli", 0.96992);
       if m25 = 0 and m26 = 0 and m27 = 0 then m28=rand("bernoulli", 0.01364);
end;

if m28=. then m29=.;
else m29 = rand("bernoulli", m36cadj);
if m29=0 then m29=.;
if m29=1 then do;
       if m27 = 0 and m28 = 1 then m29=rand("bernoulli", 0.97044);
```

```
        if m27 = 0 and m28 = 0 then m29=rand("bernoulli", 0.01757);
        if m26 = 1 and m27 = 1 and m28 = 1 then m29=rand("bernoulli", 0.99215);
        if m26 = 1 and m27 = 1 and m28 = 0 then m29=rand("bernoulli", 0.05200);
        if m26 = 0 and m27 = 1 and m28 = 1 then m29=rand("bernoulli", 0.98699);
        if m26 = 0 and m27 = 1 and m28 = 0 then m29=rand("bernoulli", 0.04022);
        if m26 = 0 and m27 = 0 and m28 = 1 then m29=rand("bernoulli", 0.96992);
        if m26 = 0 and m27 = 0 and m28 = 0 then m29=rand("bernoulli", 0.01364);
end;

if m29=. then m30=.;
else m30 = rand("bernoulli", m36cadj);
if m30=0 then m30=.;
if m30=1 then do;
        if m28 = 0 and m29 = 1 then m30=rand("bernoulli", 0.97044);
        if m28 = 0 and m29 = 0 then m30=rand("bernoulli", 0.01757);
        if m27 = 1 and m28 = 1 and m29 = 1 then m30=rand("bernoulli", 0.99215);
        if m27 = 1 and m28 = 1 and m29 = 0 then m30=rand("bernoulli", 0.05200);
        if m27 = 0 and m28 = 1 and m29 = 1 then m30=rand("bernoulli", 0.98699);
        if m27 = 0 and m28 = 1 and m29 = 0 then m30=rand("bernoulli", 0.04022);
        if m27 = 0 and m28 = 0 and m29 = 1 then m30=rand("bernoulli", 0.96992);
        if m27 = 0 and m28 = 0 and m29 = 0 then m30=rand("bernoulli", 0.01364);
end;

if m30=. then m31=.;
else m31 = rand("bernoulli", m36cadj);
if m31=0 then m31=.;
if m31=1 then do;
        if m29 = 0 and m30 = 1 then m31=rand("bernoulli", 0.97044);
        if m29 = 0 and m30 = 0 then m31=rand("bernoulli", 0.01757);
        if m28 = 1 and m29 = 1 and m30 = 1 then m31=rand("bernoulli", 0.99215);
        if m28 = 1 and m29 = 1 and m30 = 0 then m31=rand("bernoulli", 0.05200);
```

```
        if m28 = 0 and m29 = 1 and m30 = 1 then m31=rand("bernoulli", 0.98699);
        if m28 = 0 and m29 = 1 and m30 = 0 then m31=rand("bernoulli", 0.04022);
        if m28 = 0 and m29 = 0 and m30 = 1 then m31=rand("bernoulli", 0.96992);
        if m28 = 0 and m29 = 0 and m30 = 0 then m31=rand("bernoulli", 0.01364);
end;

if m31=. then m32=.;
else m32 = rand("bernoulli", m36cadj);
if m32=0 then m32=.;
if m32=1 then do;
        if m30 = 0 and m31 = 1 then m32=rand("bernoulli", 0.97044);
        if m30 = 0 and m31 = 0 then m32=rand("bernoulli", 0.01757);
        if m29 = 1 and m30 = 1 and m31 = 1 then m32=rand("bernoulli", 0.99215);
        if m29 = 1 and m30 = 1 and m31 = 0 then m32=rand("bernoulli", 0.05200);
        if m29 = 0 and m30 = 1 and m31 = 1 then m32=rand("bernoulli", 0.98699);
        if m29 = 0 and m30 = 1 and m31 = 0 then m32=rand("bernoulli", 0.04022);
        if m29 = 0 and m30 = 0 and m31 = 1 then m32=rand("bernoulli", 0.96992);
        if m29 = 0 and m30 = 0 and m31 = 0 then m32=rand("bernoulli", 0.01364);
end;

if m32=. then m33=.;
else m33 = rand("bernoulli", m36cadj);
if m33=0 then m33=.;
if m33=1 then do;
        if m31 = 0 and m32 = 1 then m33=rand("bernoulli", 0.97044);
        if m31 = 0 and m32 = 0 then m33=rand("bernoulli", 0.01757);
        if m30 = 1 and m31 = 1 and m32 = 1 then m33=rand("bernoulli", 0.99215);
        if m30 = 1 and m31 = 1 and m32 = 0 then m33=rand("bernoulli", 0.05200);
        if m30 = 0 and m31 = 1 and m32 = 1 then m33=rand("bernoulli", 0.98699);
        if m30 = 0 and m31 = 1 and m32 = 0 then m33=rand("bernoulli", 0.04022);
        if m30 = 0 and m31 = 0 and m32 = 1 then m33=rand("bernoulli", 0.96992);
```

```
        if m30 = 0 and m31 = 0 and m32 = 0 then m33=rand("bernoulli", 0.01364);
end;

if m33=. then m34=.;
else m34 = rand("bernoulli", m36cadj);
if m34=0 then m34=.;
if m34=1 then do;
        if m32 = 0 and m33 = 1 then m34=rand("bernoulli", 0.97044);
        if m32 = 0 and m33 = 0 then m34=rand("bernoulli", 0.01757);
        if m31 = 1 and m32 = 1 and m33 = 1 then m34=rand("bernoulli", 0.99215);
        if m31 = 1 and m32 = 1 and m33 = 0 then m34=rand("bernoulli", 0.05200);
        if m31 = 0 and m32 = 1 and m33 = 1 then m34=rand("bernoulli", 0.98699);
        if m31 = 0 and m32 = 1 and m33 = 0 then m34=rand("bernoulli", 0.04022);
        if m31 = 0 and m32 = 0 and m33 = 1 then m34=rand("bernoulli", 0.96992);
        if m31 = 0 and m32 = 0 and m33 = 0 then m34=rand("bernoulli", 0.01364);
end;

if m34=. then m35=.;
else m35 = rand("bernoulli", m36cadj);
if m35=0 then m35=.;
if m35=1 then do;
        if m33 = 0 and m34 = 1 then m35=rand("bernoulli", 0.97044);
        if m33 = 0 and m34 = 0 then m35=rand("bernoulli", 0.01757);
        if m32 = 1 and m33 = 1 and m34 = 1 then m35=rand("bernoulli", 0.99215);
        if m32 = 1 and m33 = 1 and m34 = 0 then m35=rand("bernoulli", 0.05200);
        if m32 = 0 and m33 = 1 and m34 = 1 then m35=rand("bernoulli", 0.98699);
        if m32 = 0 and m33 = 1 and m34 = 0 then m35=rand("bernoulli", 0.04022);
        if m32 = 0 and m33 = 0 and m34 = 1 then m35=rand("bernoulli", 0.96992);
        if m32 = 0 and m33 = 0 and m34 = 0 then m35=rand("bernoulli", 0.01364);
end;
```

```
        if m35=. then m36=.;
        else m36 = rand("bernoulli", m36cadj);
        if m36=0 then m36=.;
        if m36=1 then do;
               if m34 = 0 and m35 = 1 then m36=rand("bernoulli", 0.97044);
               if m34 = 0 and m35 = 0 then m36=rand("bernoulli", 0.01757);
               if m33 = 1 and m34 = 1 and m35 = 1 then m36=rand("bernoulli", 0.99215);
               if m33 = 1 and m34 = 1 and m35 = 0 then m36=rand("bernoulli", 0.05200);
               if m33 = 0 and m34 = 1 and m35 = 1 then m36=rand("bernoulli", 0.98699);
               if m33 = 0 and m34 = 1 and m35 = 0 then m36=rand("bernoulli", 0.04022);
               if m33 = 0 and m34 = 0 and m35 = 1 then m36=rand("bernoulli", 0.96992);
               if m33 = 0 and m34 = 0 and m35 = 0 then m36=rand("bernoulli", 0.01364);
        end;
     group='A';
     output;
     end;
run;

data VNS_mediumCORE;
set VNS_mediumCORE;
total12=sum(of m1-m12);
total24=sum(of m1-m24);
total36=sum(of m1-m36);
if total12=. then total12=0;
if total24=. then total24=0;
if total36=. then total36=0;
run;

data temp12;
set VNS_mediumCORE;
if total12=0 then output;
```

```
array m [*] m1-m12;
     if m[1] = 1 then do; start=1; output; end;
     do i=2 to dim(m);
          if m[i] = 1 and m[i-1] = 0 then do; start=i; output; end;
     end;
     if m[12] = 1 then do; end=12; output; end;
     do i=1 to dim(m)-1;
          if m[dim(m)-i] = 1 and (m[dim(m)-i+1] = 0 or m[dim(m)-i+1] = .) then do; end=dim(m)-i;
output; end;
     end;
keep x start end;
run;

data start12;
set temp12;
if end=.;
keep x start;
data end12;
set temp12;
if start=. or end^=.;
keep x end;
run;

proc sort data=end12;
by x end;
run;

data dur12;
set start12;
set end12;
dur12=end-start+1;
```

```
keep x dur12;
run;

proc sort data=dur12;
by x descending dur12;
run;

proc sort data=dur12 nodupkey;
by x;
run;


data temp24;
set VNS_mediumCORE;
if total24=0 then output;
array m [*] m1-m24;
     if m[1] = 1 then do; start=1; output; end;
     do i=2 to dim(m);
          if m[i] = 1 and m[i-1] = 0 then do; start=i; output; end;
     end;
     if m[24] = 1 then do; end=24; output; end;
     do i=1 to dim(m)-1;
          if m[dim(m)-i] = 1 and (m[dim(m)-i+1] = 0 or m[dim(m)-i+1] = .) then do; end=dim(m)-i;
output; end;
     end;
keep x start end;
run;

data start24;
set temp24;
if end=.;
```

```
keep x start;
data end24;
set temp24;
if start=. or end^=.;
keep x end;
run;

proc sort data=end24;
by x end;
run;

data dur24;
set start24;
set end24;
dur24=end-start+1;
keep x dur24;
run;

proc sort data=dur24;
by x descending dur24;
run;

proc sort data=dur24 nodupkey;
by x;
run;


data temp36;
set VNS_mediumCORE;
if total36=0 then output;
array m [*] m1-m36;
```

```
      if m[1] = 1 then do; start=1; output; end;
      do i=2 to dim(m);
           if m[i] = 1 and m[i-1] = 0 then do; start=i; output; end;
      end;
      if m[36] = 1 then do; end=36; output; end;
      do i=1 to dim(m)-1;
           if m[dim(m)-i] = 1 and (m[dim(m)-i+1] = 0 or m[dim(m)-i+1] = .) then do; end=dim(m)-i;
output; end;
      end;
keep x start end;
run;

data start36;
set temp36;
if end=.;
keep x start;
data end36;
set temp36;
if start=. or end^=.;
keep x end;
run;

proc sort data=end36;
by x end;
run;

data dur36;
set start36;
set end36;
dur36=end-start+1;
keep x dur36;
```

```
run;

proc sort data=dur36;
by x descending dur36;
run;

proc sort data=dur36 nodupkey;
by x;
run;

data pop.VNS_mediumCORE;
set VNS_mediumCORE;
set dur12;
set dur24;
set dur36;
if total12=0 then dur12=0;
if total24=0 then dur24=0;
if total36=0 then dur36=0;
run;

ods pdf file="S:\Rep_Client_Project_current\Simpop\VNS_pop_desc.pdf";
title "Frequency of missings, non-responders and responders in VNS population";
proc freq data=pop.VNS_mediumCORE;
table m1-m36/missing;
run;

title "Distribution of duration of response in VNS population";
proc univariate data=pop.VNS_mediumCORE;
var dur36;
run;
```

```
ods pdf close;

proc univariate data=pop.VNS_mediumCORE;
var dur12 total12 dur24 total24 dur36 total36;
run;

proc datasets lib=work nolist memtype=data kill;
run;
quit;
```