Primerjava opisnih spremenljivk, logistična regresija (8**)

Seminarska naloga pri predmetu Osnove teoretične statistike

Urh Peček in Maja Urankar

2020/21

Kazalo

Navodilo naloge

Zdravniki želijo primerjati dve vrsti operacije, imajo nek vzorec bolnikov (cca 300 bolnikov) z rakom, pri polovici bolnikov je bila narejena ena vrsta operacije, pri drugi polovici druga. Zanimata jih dve vprašanji:

- Ali je bila odločitev za vrsto operacije povezana s stadijem bolnika (vsak bolnik je v enem od štirih stadijev bolezni)?
- Ali ena vrsta operacije povzroča manj zapletov kot druga?

Ker vsi zapleti niso enakovredni, primerjajo vsakega posebej, zanima jih 10 različnih zapletov. Naredijo 10 primerjav (za vsakega bolnika pogledajo ali se mu je nek zaplet zgodil ali ne). Pri preverjanju povezanosti s stadijem naredijo enako - 4 primerjave, za vsak stadij posebej torej primerjajo dve opisni spremenljivki (je oz. ni v stadiju in prva/druga vrsta operacije).

1. Oglejte si, kaj se zgodi z napako I. reda pri takem načinu preverjanja. Komentirajte.
2. Podajte kak predlog, kako bi tovrstne podatke lahko ustrezneje analizirali.

Uvod

Odgovoriti želimo na dve raziskovalni vprašanji in v ta namen si postavimo dve ničelni domnevi, ki ju bomo v seminarski nalogi preverjali. Ena od domnev je povezana z zapleti, druga pa s stadiji.

Domnevi, ki pripadata zapletom, sta:

H_0: Operacija 1 povzroča enako število zapletov kot operacija 2.

H_A: Operaciji se razlikujeta v številu zapletov.

Domnevi, ki pripadata stadijem, sta:

H_0: Operacija ni povezana s stadijem bolnika.

H_A: Operacija je povezana s stadijem bolnika.

Da bi lahko odgovorili na raziskovalni vprašanji, bomo ustvarili več testov in preverili, kateri je najustreznejši ter podali priporočila glede njihove uporabe. Kot stopnjo statistične značilnosti bomo upoštevali vrednost $\alpha = 0.05$.

Logistična regresija

V nadaljevanju bomo uporabili tudi logistično regresijo, ki je tekom študija še nismo spoznali, zato podajmo nekaj teorije.

Logistično regresijo uporabljamo, kadar imamo v modelu eno ali več kategoričnih ali številskih pojasnjevalnih spremenljivk, odvisna spremenljivka pa je kategorična, najpogosteje je binarna (0/1). V našem primeru sta možna izida odvisne spremenljivke: 1. operacija (0) ali 2. operacija (1).

Logistični model lahko v linearni obliki zapišemo kot:

$$\ln(\frac{P}{1-P}) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k$$

kjer je $\frac{P}{1-P}$ razmerje obetov, β ocenjeni parametri v modelu ter x pojasnjevalne spremenljivke v modelu.

Interpretacija parametrov v logistični regresiji:

Parametri v logistični regresiji so ocenjeni po metodi največjega verjetja. Interpretacija je pri logistični regresiji podobna interpretaciji parametrov v linearni regresiji, le da parameter β predstavlja spremembo logaritma razmerja obetov, če se pojasnjevalna spremenljivka poveča za eno enoto (v primeru multiple logistične regresije moramo biti pozorni, da parameter interpretiramo glede na ostale pojasnjevalne spremenljivke v modelu). Če je parameter β pozitiven, je razmerje obetov (e^{β}) večje od 1, torej je bolj verjeten izid 1 (oz. 2. operacija v našem primeru), obratno velja, če je parameter β negativen, je razmerje obetov manjše od 1, kar pa pomeni, da je bolj verjeten izid 0 (v našem primeru 1. operacija).

Generiranje zapletov in stadijev

Primarni del, ki nam omogoča testiranje domnev, je podatkovni okvir, ki za vsakega bolnika vsebuje informacije o operaciji ter zapletih oz. stadiju. Tako smo kot prvo definirali funkciji, ki nam za 300 bolnikov, od katerih jih je 150 na operaciji 1 in 150 na operaciji 2, priredita pojav zapleta oz. pripadnost stadiju.

Zapleti

Zaplete smo ustvarili s funkcijo `narediZaplet(p)`, ki za vsakega bolnika in vsak zaplet simulira ali se je zaplet zgodil (1) ali se zaplet ni zgodil (0). V vektorju $p = (p_1, ..., p_{20})$ je p_i za $i = 2n + 1$ verjetnost, da se i-ti zaplet pojavi pri operaciji 1, medtem, ko je p_i za $i = 2n$ verjetnost, da se i-ti zaplet pojavi pri operaciji 2.

Tabela 1: Primer podatkov za zaplete

	operacija	zaplet1	zaplet2	zaplet3	zaplet4	zaplet5	zaplet6	zaplet7	zaplet8	zaplet9	zaplet10
146	0	0	0	1	0	0	0	0	1	0	1
147	0	0	0	1	1	0	1	1	1	0	0
148	0	0	0	1	1	1	1	1	1	0	1
149	0	0	0	0	0	0	0	0	0	0	0
150	0	0	1	1	1	0	1	1	1	0	1
151	1	0	0	1	0	0	1	0	0	1	0
152	1	1	0	1	0	1	1	1	1	0	0
153	1	1	1	0	0	1	1	1	0	1	1
154	1	1	0	0	1	1	1	1	1	1	0
155	1	1	1	0	0	0	1	1	0	0	0

Zapleti z odvisnostjo

Ker je verjetnost zapletov v praksi lahko odvisna tudi od bolnikovih lastnosti, smo bolnike poleg tega, da jih 150 pripada prvi operaciji in 150 drugi operaciji, ločili še v dve dodatni skupini (0/1), ki pogojujeta verjetnost posameznega zapleta pri obeh operacijah ter vseh zapletih z enakim faktorjem. S tem smo ustvarili najbolj primitivno odvisnost verjetnosti zapleta. Funkcija `narediZapletOdvisnost(p_odvisnost,p)` za vsakega bolnika in vsak zaplet simulira ali se je zaplet zgodil (1) ali se zaplet ni zgodil (0).

V vektorju $p = (p_1, ..., p_{20})$ je p_i za $i = 2n + 1$ začetna verjetnost, da se i-ti zaplet pojavi pri operaciji 1, medtem, ko je p_i za $i = 2n$ začetna verjetnost, da se i-ti zaplet pojavi pri operaciji 2. Objekt `p_odvisnost` je numerični vektor dolžine ena znotraj intervala [0, inf), s katerim zmanjšamo (< 1) ali povečamo (> 1) verjetnost posameznega zapleta pri vsaki operaciji. Sprememba verjetnosti zapleta je odvisna od bolnikove pripadnosti skupini. Če bolnik pripada skupini 0, je verjetnost zapleta enaka začetni, sicer pa se verjetnost zapleta pomnoži z argumentom `p_odvisnost` pri čemer verjetnost seveda ostane na intervalu [0,1].

Stadiji

Stadije smo ustvarili s funkcijo `narediStadij(p)`, ki za podane verjetnosti $p = (p_1, ..., p_8)$ vsakemu bolniku priredi natanko en stadij (dotičnemu stadiju je dodeljena 1, ostalim stadijem pa 0). Bolnike ponovno ločimo na dva dela, 150 bolnikov, ki so imeli operacijo 1 in 150 bolnikov, ki so imeli operacijo 2. V vektorju je p_i za $i = 1, ..., 4$ verjetnost, da ima bolnik pri operaciji 1 i-ti stadij in p_i za $i = 5, ..., 8$ verjetnost, da ima bolnik pri operaciji 2 i-ti stadij.

Tabela 2: Primer podatkov za stadije

	operacija	stadij	stadij1	stadij2	stadij3	stadij4
146	0	2	0	1	0	0
147	0	2	0	1	0	0

	operacija	stadij	stadij1	stadij2	stadij3	stadij4
148	0	1	1	0	0	0
149	0	1	1	0	0	0
150	0	4	0	0	0	1
151	1	1	1	0	0	0
152	1	4	0	0	0	1
153	1	3	0	0	1	0
154	1	2	0	1	0	0
155	1	4	0	0	0	1

Verjetnost napake I. vrste brez popravka

Preverimo kako je z verjetnostjo napake 1. vrste ob testiranju več domnev hkrati, pri čemer se ne poslužimo nobenega izmed popravkov.

Simulacija zapletov pod ničelno domnevo

V namen izračuna verjetnosti napake 1. vrste brez popravka pri testiranju domneve o zapletih, definiramo funkcijo `simulacija_zapleti_H0(p)`, ki vrne vrednost 1, če je vsaj en zaplet statistično značilno povezan z vrsto operacije in vrednost 0, če noben od zapletov ni statistično značilno povezan z vrsto operacije. Za natančnejši izračun, s katerim zmanjšamo odstopanje simulirane vrednosti od prave, definiramo funkcijo `simulacija_2_zapleti_H0(p)`, ki za dan vektor verjetnosti zapletov s pomočjo zgornje funkcije izračuna verjetnost napake I. vrste pri domnevi, da nobeden od zapletov ni povezan z vrsto operacije.

Pričakujemo, da bo verjetnost napake I. vrste v primeru, ko testiramo 10 neodvisnih domnev (pri stopnji značilnosti $\alpha = 0.05$) $1 - 0.95^{10} = 0.40$. To preverimo s simulacijo.

Simulirana verjetnost napake I. vrste pri hkratnem testiranju 10 domnev, ki so neodvisne med sabo, pri vzorcu velikosti $n = 30$, kjer je v vsakem vzorcu 10.000 ponovitev, variira okrog $\overline{X} = 0.442$ z vzorčnim standardnim odklonom $\hat{\sigma} = 0.005$. 95% interval zaupanja za pravo vrednost verjetnosti napake prve vrste tako izračunamo kot $\overline{X} \pm t_{1-\frac{\alpha}{2};n-1} \cdot \hat{\sigma}/\sqrt{n}$ in je enak `[0.440,0.444]`. Vidimo, da simulirana vrednost verjetnosti napake I. vrste v povprečju odstopa od teoretično izračunane in sicer je previsoka.

Do odstopanja od teoretično izračunane verjetnosti napake I. vrste, ki je bila prvotno izračunana, pride, ker smo pri izračunu predpostavili, da je velikost testa $\alpha = 0.05$. S simulacijo smo preverili, kakšna je velikost testa v primeru, ko testiramo eno domnevo (oz. naredimo en test) in pri 200 ponovitvah 1000 simulacij je bilo ugotovljeno, da je velikost testa v povprečju 0.0571 (95% interval zaupanja: `[0.0562, 0.0582]`), torej je večja od 0.05 (ničelno domnevo zavrnemo večkrat, kot bi pričakovali). Razlog za previsoko velikost testa je ta, da je hi-kvadrat asimptotski test. S tem se spremeni tudi teoretični izračun verjetnosti napake I. vrste v primeru, ko večkratno testiramo 10 neodvisnih domnev. Pričakujemo torej, da bo v takem primeru verjetnost napake I. vrste $1 - (1 - 0.0571)^{10} = 0.444$, kar se sklada z rezultatom iz simulacij.

Simulacija stadijev pod ničelno domnevo

V namen izračuna verjetnosti napake 1. vrste brez popravka pri testiranju domneve o stadijih definiramo funkcijo `simulacija_stadiji_H0(p)`, ki vrne vrednost 1, če je vsaj en stadij povezan z vrsto operacije in vrednost 0, če noben stadij ni povezan z vrsto operacije. Pri tem za vsak stadij $i = 1, ..., 4$ s hi-kvadrat testom (brez Yatesovega popravka) testira ničelno domnevo, da i-ti stadij in vrsta operacije nista povezana. Za natančnejši izračun, s katerim zmanjšamo odstopanje simulirane vrednosti od prave, definiramo funkcijo `simulacija_2_stadiji_H0(p)`, ki za dan vektor verjetnosti stadijev s pomočjo zgornje funkcije izračuna verjetnost napake I. vrste pri domnevi, da nobeden od stadijev ni povezan z vrsto operacije.

V primeru neodvisnih spremenljivk bi na podlagi produkta velikosti testov pričakovali, da bo verjetnost napake I. vrste v primeru, ko testiramo 4 domneve (pri stopnji značilnosti $\alpha = 0.05$), $1 - 0.95^4 = 0.19$.

V našem primeru pa so spremenljivke in posledično velikosti testov odvisne, saj je pri vsakem bolniku vsota verjetnosti prisotnosti stadijev enaka 1 in zato ne moremo pričakovati, da bo verjetnost napake I. vrste enaka 0.19. Verjetnost napake I. vrste bo zaradi odvisnosti manjša. S simulacijo preverimo, kakšne velikosti je v resnici pri testiranju štirih domnev, ki niso neodvisne.

Simulirana vrednost verjetnosti napake I. vrste pri hkratnem testiranju 4 domnev, ki med seboj niso neodvisne, pri vzorcu velikosti $n = 30$, kjer je v vsakem vzorcu 10.000 ponovitev variira okrog $\overline{X} = 0.170$ z vzorčnim standardnim odklonom $\hat{\sigma} = 0.003$. 95% interval zaupanja za pravo vrednost verjetnosti napake prve vrste tako izračunamo kot $\overline{X} \pm t_{1-\frac{\alpha}{2};n-1} \cdot \hat{\sigma}/\sqrt{n}$ in je enak `[0.169,0.171]`. S tem smo pokazali, da odstopanje ni naključno in da je velikost testa manjša od teoretično izračunane. Razlog je v tem, da smo v teoriji predpostavili, da testiramo 4 neodvisne domneve, v praksi pa so stadiji in posledično testi med seboj odvisni.

Statistični testi

Spodaj bomo opisali statistične teste in uporabljene simulacije za izračun njihovih moči oziroma velikosti, s katerimi smo testirali domneve o zapletih oziroma stadijih. Psevdokode opisanih testov in njihovih simulacij najdemo pod poglavjem Psevdokoda.

Zapleti

Osnovni test

Definiramo funkcijo `simulacija_zapleti_smeri(p)`, ki za dan vektor verjetnosti zapletov za vsak zaplet $i = 1, \ldots, 10$ s hi-kvadrat testom (brez Yatesovega popravka) testira ničelno domnevo, da i-ti zaplet in vrsta operacije nista povezana. Funkcija z upoštevanjem Bonferronijevega popravka pove, če je število zapletov statistično značilno odvisno od operacije in vrsto operacije pri kateri se zgodi več zapletov.

Opomba: Rezultat funkcije je potrebno jemati nekoliko "z rezervo" saj nam vrne vrednost FALSE, torej, da ne zavrnemo domneve, da število zapletov ni povezano z vrsto operacije v primeru, ko je vsota zapletov, ki so statistično značilno bolj pogosti pri eni operaciji enaka vsoti zapletov, ki so statistično značilno bolj pogosti pri drugi operaciji, kar pa v bistvu ne pomeni nujno, da je število zapletov pri obeh operacijah približno enako, saj do statistične značilnosti lahko pride pri različni absolutni razliki med številom zapletov posameznega zapleta med operacijama.

Nadalje definiramo funkcijo `simulacija_2_zapleti(p)`, ki s pomočjo zgornje funkcije vrne simulirano moč oz. velikost testa in pove, pri kateri operaciji se pojavi več zapletov.

Podobno definiramo tudi funkciji `simulacija_zapleti_smeri_odvisnost(p_odvisnost,p)` in `simulacija_2_zapleti_smeri_odvisnost(p_odvisnost,p)`, ki vrne simulirano moč oz. velikost testa z vključeno odvisnostjo zapletov in pove, pri kateri operaciji se pojavi več zapletov.

Logistično seštevanje

Definiramo funkcijo `simulacija_zapleti_logisticna_sum(p)`, ki za dan vektor verjetnosti zapletov naredi logistični model znotraj katerega s pomočjo seštevanja zapletov testiramo domnevo, da število zapletov ni povezano z vrsto operacije. Funkcija nam pove, če se število zapletov pri operacijah statistično značilno razlikuje.

Nadalje definiramo funkcijo `simulacija_2_zapleti_logisticna_sum(p)`, ki s pomočjo zgornje funkcije vrne simulirano moč oziroma velikost testa in pove, pri kateri operaciji so zapleti statistično značilno bolj pogosti.

Podobno definiramo tudi funkciji `simulacija_zapleti_logisticna_sum_odvisnost(p_odvisnost,p)` in `simulacija_2_zapleti_logisticna_sum_odvisnost(p_odvisnost,p)`, ki vrne simulirano moč oz. velikost testa z vključeno odvisnostjo zapletov in pove, pri kateri operaciji so zapleti statistično značilno bolj pogosti.

Logistično večkratno testiranje

Definiramo funkcijo `simulacija_zapleti_logisticna(p)`, ki za dan vektor verjetnosti zapletov naredi logistični model, znotraj katerega testiramo 10 ničelnih domnev, da zaplet $i = 1, ..., 10$ ni povezan z vrsto operacije. Funkcija nam pove, če se število zapletov pri operacijah statistično značilno razlikuje ter operacijo, pri kateri se pojavi več zapletov.

Opomba: Rezultat funkcije je potrebno jemati nekoliko "z rezervo" saj nam vrne vrednost FALSE, torej, da ne zavrnemo domneve, da število zapletov ni povezano z vrsto operacije v primeru, ko je vsota zapletov, ki so statistično značilno bolj pogosti pri eni operaciji enaka vsoti zapletov, ki so statistično značilno bolj pogosti pri drugi operaciji, kar pa v bistvu ne pomeni nujno, da je število zapletov pri obeh operacijah približno enako, saj do statistične značilnosti lahko pride pri različni absolutni razliki med številom zapletov posameznega zapleta med operacijama.

Nadalje definiramo funkcijo `simulacija_2_zapleti_logisticna(p)`, ki s pomočjo zgornje funkcije vrne simulirano moč oz. velikost testa in pove, pri kateri operaciji so zapleti statistično značilno bolj pogosti.

Podobno definiramo tudi funkciji `simulacija_zapleti_logisticna_odvisnost(p_odvisnost,p)` in `simulacija_2_zapleti_logisticna_odvisnost(p_odvisnost,p)`, ki vrne simulirano moč oz. velikost testa z vključeno odvisnostjo zapletov in pove, če se število zapletov statistično značilno razlikuje, ter operacijo pri kateri so zapleti statistično značilno bolj pogosti.

Stadiji

Hi-kvadrat test

Definiramo funkcijo `simulacija_stadiji_hi2(p)`, ki za dan vektor verjetnosti stadijev s hi-kvadrat testom preizkusi domnevo, da stadiji niso povezani z vrsto operacije. Testiranje poteka za vse stadije naenkrat tj. povezanosti posameznega stadija z operacijo ne testiramo posebej. Funkcija nam pove ali ničelno domnevo zavrnemo ali jo obdržimo.

Nadalje definiramo funkcijo `simulacija_2_stadiji_hi2(p)`, ki nam s pomočjo zgornje funkcije vrne simulirano moč oz. velikost testa.

Velja, da so pri vsakem bolniku stadiji popolno linearno odvisni tj. četrti stadij je linearna kombinacija ostalih treh. To v splošnem vpliva na velikost oz. moč testa, v zgornjem primeru pa uporabimo test hi-kvadrat, ki temelji na kontingenčni tabeli, kjer pa je ena celica popolna linearna kombinacija ostalih, ne glede na odvisnost oz. neodvisnost spremenljivk v podatkih. Tako je test hi-kvadrat primeren tudi za odvisne podatke in s tem za testiranje naše ničelne domneve.

Osnovni test

Definiramo funkcijo `simulacija_stadiji(p)`, ki za dan vektor verjetnosti stadijev za vsak stadij $i = 1, ..., 4$ s hi-kvadrat testom (brez Yatesovega popravka) testira ničelno domnevo, da i-ti stadij in vrsta operacije nista povezana. Za vsak stadij določimo operacijo, pri kateri je stadij statistično značilno bolj pogost (0/1/2).

Nadalje definiramo funkcijo `simulacija_2_stadiji`, ki s pomočjo zgornje funkcije vrne simulirano moč oz. velikost testa in za vsak stadij pove, katera operacija je bolj pogosta.

Opomba: V postopku simulacije je uporabljen studentov test t, ki predpostavlja, da so podatki obeh skupin porazdeljeni normalno in da sta varianci med skupinama vsaj približno (pri določeni stopnji značilnosti) enaki. V našem primeru so podatki diskretni tj. celoštevilski in se včasih lahko zgodi, da test ne deluje, saj predpostavke niso izpolnjene. To predstavlja težavo zgolj pri simulacijah moči in velikosti pri različnih ničelnih in alternativnih "domnevah" saj za nekatere verjetnosti stadijev ni mogoče preveriti moči oziroma velikosti, ne pa tudi pri kasnejšem resničnem preverjanju ničelnih domnev, kjer zgornje simulacije v bistvu ne uporabimo. Te težave se zavedamo, vendar pa se načeloma moči oziroma velikosti nebi smele tako močno "sekati", da bi prišlo do večjih napak pri kasnejši izbiri testa. Problem pri izvedbi izračuna v bistvu nastane, ko so preverjane verjetnosti za isti stadij tako različne, da v vsaki iteraciji simulacije dobimo nagnjenost k isti operaciji in posledično je izvedba t-testa nemogoča.

Pri zgornjem testu preverjamo domnevo o ničelni povezanosti posameznega stadija z vrsto operacije, za kar uporabimo štiri hi-kvadrat teste. Kot smo že videli, to vodi v previsoko verjetnost napake I. vrste. Te težave smo se v našem primeru rešili z uporabo Bonferronijevega popravka, ki pa je v svojem smislu precej strog, kar pomeni, da zniža velikost oz. moč testa. Dodatno pa moč zniža še odvisnost testov, ki izhaja iz odvisnosti stadijev. Test je tako precej konservativen.

Logistično testiranje

Definiramo funkcijo `simulacija_stadiji_logisticna(p)`, ki za dan vektor verjetnosti naredi logistični model, znotraj katerega testiramo 4 ničelne domneve, da stadij $i = 1, ..., 4$ ni povezan z vrsto operacije. Za vsak stadij določimo operacijo, pri kateri je stadij statistično značilno bolj pogost (0/1/2).

Nadalje definiramo funkcijo `simulacija_2_stadiji_logisticna(p)`, ki s pomočjo zgornje funkcije vrne simulirano moč oz. velikost testa in za vsak stadij pove, pri kateri operaciji je bil stadij statistično značilno bolj pogost.

Opomba: V postopku simulacije je uporabljen studentov test t, ki predpostavlja, da so podatki obeh skupin porazdeljeni normalno in da sta varianci med skupinama vsaj približno (pri določeni stopnji značilnosti) enaki. V našem primeru so podatki diskretni tj. celoštevilski in se včasih lahko zgodi, da test ne deluje, saj predpostavke niso izpolnjene. To predstavlja težavo zgolj pri simulacijah moči in velikosti pri različnih ničelnih in alternativnih "domnevah" saj za nekatere verjetnosti stadijev ni mogoče preveriti moči oziroma velikosti, ne pa tudi pri kasnejšem resničnem preverjanju ničelnih domnev, kjer zgornje simulacije v bistvu ne uporabimo. Te težave se zavedamo, vendar pa se načeloma moči oziroma velikosti nebi smele tako močno "sekati", da bi prišlo do večjih napak pri kasnejši izbiri testa. Problem pri izvedbi izračuna v bistvu nastane, ko so preverjane verjetnosti za isti stadij tako različne, da v vsaki iteraciji simulacije dobimo nagnjenost k isti operaciji in posledično je izvedba t-testa nemogoča.

Pri testu z logistično regresijo v splošnem preverjamo štiri ničelne domneve ($\mu_1 = 0$; $\mu_2 - \mu_1 = 0$; $\mu_3 - \mu_1 = 0$; $\mu_4 - \mu_1 = 0$), ki so med seboj odvisne, zato je za ustrezne vrednosti p, ki pripadajo domnevam, ki testirajo vrednosti parametrov, potrebno upoštevati odvisnost hkratnih primerjav. To dosežemo s funkcijo `glht` iz paketa `multcomp`, ki temelji na multivariatni t porazdelitvi. V našem primeru smo ničelne domneve prilagodili, tako da testirajo posamezne vrednosti parametrov in ne razlik glede na referenčno skupino. Testiramo torej domneve $\mu_1 = 0$, $\mu_2 = 0$, $\mu_3 = 0$ ter $\mu_4 = 0$. V ta namen je bila uporabljena funkcija glht, ki upošteva hkratnost primerjav, kar vodi v povečanje vrednosti p in posledično zmanjšanje moči testa.

Rezultati simulacij

Zapleti

Velikosti testov

Najprej si poglejmo kako je z velikostjo ustvarjenih testov. Pogledali so bomo velikost testov pod različnimi ničelnimi "domnevami" o zapletih in jih primerjali med seboj. Dodali bomo tudi zaplete, ki se zgodijo v odvisnosti od skupine kateri bolnik pripada. Upoštevali bomo, da je verjetnost, da bolnik pripada skupini, ki je bolj dovzetna za zaplete enaka 0.7 in se začetna verjetnost zapleta pri bolnikih v skupini, ki ni dovzetna za zaplete pomnoži z 1/3.

0.1) `p_zaplet_H0_1 = rep(0.5,20)`

V vzorcu zapletov ne vidimo vzorca v smislu pogostosti zapletov in njihove povezanosti z operacijo. Vsak zaplet se zgodi pri približno vsakem drugem bolniku, pri obeh operacijah enako.

Tabela 3: Simulacija za zaplete H0_1

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Velikost	0.048	0.036	0.049
Velikost [odvisnost]	0.047	0.032	0.039

0.2) `p_zaplet_H0_2 = rep(0.2,20)`

V vzorcu zapletov ne vidimo vzorca v smislu pogostosti zapletov in njihove povezanosti z operacijo. Vsak zaplet se zgodi pri približno vsakem petem bolniku, pri obeh operacijah enako.

Tabela 4: Simulacija za zaplete H0_2

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Velikost	0.049	0.037	0.042
Velikost [odvisnost]	0.044	0.031	0.037

0.3) `p_zaplet_H0_2 = rep(0.8,20)`

V vzorcu zapletov ne vidimo vzorca v smislu pogostosti zapletov in njihove povezanosti z operacijo. Vsak zaplet se zgodi približno na vsake 4 izmed 5 bolnikov, pri obeh operacijah enako.

Tabela 5: Simulacija za zaplete H0_3

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Velikost	0.046	0.047	0.040
Velikost [odvisnost]	0.047	0.041	0.041

Moči testov

Sedaj pa si poglejmo kako je z močjo ustvarjenih treh testov. Pogledali so bomo velikost testov pod različnimi alternativnimi "domnevami" o zapletih in jih primerjali med seboj.

1.1) `p_zaplet_HA_1 = c(0.9,0.8,rep(0.15,18))`

V vzorcu zapletov vidimo zelo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavlja se večinoma zgolj eden izmed zapletov (1) in je zelo pogost, nekje na 4 od 5 bolnikov, za odtenek bolj pogost je sicer pri prvi operaciji. Ostali zapleti so zelo redki.

Tabela 6: Simulacija za zaplete HA_1

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.391	0.072	0.345
Moč [odvisnost]	0.066	0.047	0.046

1.2) `p_zaplet_HA_2 = c(0.9,0.7,rep(0.15,18))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavlja se večinoma zgolj eden izmed zapletov (1) in je zelo pogost, nekje na 4 od 5 bolnikov, nekoliko bolj pogost je sicer pri prvi operaciji. Ostali zapleti so zelo redki.

Tabela 7: Simulacija za zaplete HA_2

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.877	0.217	0.924
Moč [odvisnost]	0.209	0.083	0.177

1.3) `p_zaplet_HA_3 <- c(0.9,0.6,rep(0.15,18))`

V vzorcu zapletov vidimo vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavlja se večinoma zgolj eden izmed zapletov (1) in je zelo pogost, nekje na 4 od 5 bolnikov, bolj pogost je sicer pri prvi operaciji. Ostali zapleti so zelo redki.

Tabela 8: Simulacija za zaplete HA_3

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.974	0.562	0.983
Moč [odvisnost]	0.529	0.161	0.510

1.4) `p_zaplet_HA_4 = c(0.9,0.5,rep(0.15,18))`

V vzorcu zapletov vidimo vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavlja se večinoma zgolj eden izmed zapletov (1) in je zelo pogost, precej bolj pogost je sicer pri prvi operaciji, kjer se zgodi skoraj pri vsakem bolniku, pri drugi operaciji pa nekje pri vsakem drugem. Ostali zapleti so zelo redki.

Tabela 9: Simulacija za zaplete HA_4

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.846	0.772	0.981
Moč [odvisnost]	0.840	0.287	0.831

1.5) `p_zaplet_HA_5 = c(0.25,0.5,rep(0.15,18))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavlja se večinoma zgolj eden izmed zapletov (1), vendar ni prav pogost, nekoliko bolj pogost je sicer pri drugi operaciji, kjer se pojavi pri vsakem drugem bolniku. Ostali zapleti so zelo redki.

Tabela 10: Simulacija za zaplete HA_5

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.823	0.431	0.938
Moč [odvisnost]	0.547	0.144	0.517

1.6) `p_zaplet_HA_6 = c(0.9,0.8,0.3,0.4,rep(0.15,16))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavljata se večinoma zgolj dva izmed zapletov (1 in 2), pri čemer je en zaplet zelo pogost in se pojavi pri skoraj vsakem bolniku, rahlo večkrat pri prvi operaciji, drugi zaplet pa se pojavi nekoliko manj pogosto, na približno vsakega tretjega bolnika, rahlo večkrat pri drugi operaciji. Ostali zapleti so zelo redki.

Tabela 11: Simulacija za zaplete HA_6

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.367	0.327	0.402
Moč [odvisnost]	0.106	0.031	0.096

1.7) `p_zaplet_HA_7 = c(0.9,0.8,0.4,0.3,rep(0.15,16))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavljata se večinoma zgolj dva izmed zapletov (1 in 2), pri čemer je en zaplet zelo pogost in se pojavi pri skoraj vsakem bolniku, rahlo večkrat pri prvi operaciji, drugi zaplet pa se pojavi nekoliko manj pogosto, na približno vsakega tretjega bolnika, prav tako rahlo večkrat pri prvi operaciji. Ostali zapleti so zelo redki.

Tabela 12: Simulacija za zaplete HA_7

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.385	0.263	0.445
Moč [odvisnost]	0.107	0.087	0.073

1.8) `p_zaplet_HA_8 = c(0.9,0.6,0.1,0.4,rep(0.15,16))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavljata se večinoma zgolj dva izmed zapletov (1 in 2), pri čemer je en zaplet zelo pogost in se pojavi pri skoraj vsakem bolniku, večkrat sicer pri prvi operaciji, drugi zaplet pa se pojavi večinoma zgolj pri drugi operaciji, na približno vsakega drugega bolnika. Ostali zapleti so zelo redki.

Tabela 13: Simulacija za zaplete HA_8

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.040	0.029	0.041

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč [odvisnost]	0.505	0.023	12.000

1.9) `p_zaplet_HA_9 = c(0.9,0.75,0.25,0.4,rep(0.15,16))`]

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavljata se večinoma zgolj dva izmed zapletov (1 in 2), pri čemer je en zaplet zelo pogost in se pojavi pri skoraj vsakem bolniku, večkrat sicer pri prvi operaciji, drugi zaplet pa se pojavi bolj redko in pogosteje pri drugi operaciji, na približno vsakega drugega bolnika. Ostali zapleti so zelo redki.

Tabela 14: Simulacija za zaplete HA_9

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.525	0.373	0.521
Moč [odvisnost]	0.238	0.027	0.263

1.10) `p_zaplet_HA_10 = c(0.9,0.7,0.7,0.8,rep(0.15,16))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavljata se večinoma zgolj dva izmed zapletov (1 in 2), pri čemer sta si pogostosti zapletov precej enakovredni in se pojavita pri skoraj vsakem bolniku, eden od zapletov se nekoliko večkrat sicer pojavi pri prvi operaciji, drugi pa se za odtenek pogosteje pri drugi operaciji. Ostali zapleti so zelo redki.

Tabela 15: Simulacija za zaplete HA_10

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.751	0.085	0.765
Moč [odvisnost]	0.226	0.042	0.257

1.11) `p_zaplet_HA_11 = c(0.8,0.6,0.7,0.6,rep(0.25,16))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Pojavljata se večinoma zgolj dva izmed zapletov (1 in 2), pri čemer sta si pogostosti zapletov precej enakovredni in se pojavita pri približno štirih izmed pet bolnikov, oba zapleta se sicer nekoliko večkrat pojavita pri eni izmed operaciji. Ostali zapleti so bolj redki.

Tabela 16: Simulacija za zaplete HA_11

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.867	0.389	0.859
Moč [odvisnost]	0.234	0.093	0.193

1.12) `p_zaplet_HA_12 = c(0.6,0.4,0.3,0.4,rep(0.35,16))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Vsi zapleti se pojavljajo precej enako pogosto, morda se malo pogosteje pojavlja eden od zapletov, pa še to zgolj pri eni od operacij. Sicer pa se vsi ostali zapleti pojavljajo pri približno vsakem tretjem bolniku, pri obeh operacijah enako.

Tabela 17: Simulacija za zaplete HA_12

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.665	0.061	0.656
Moč [odvisnost]	0.293	0.036	0.267

1.13) `p_zaplet_HA_13 =  c(rep(c(0.40,0.50),3),rep(0.3,14))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Vsi zapleti se pojavljajo precej enako pogosto, nekoliko pogosteje se pojavljajo trije zapleti in sicer rahlo bolj pogosto pri drugi operaciji kot pri prvi. Sicer pa se vsi ostali zapleti pojavljajo pri približno vsakem tretjem bolniku, pri obeh operacijah enako.

Tabela 18: Simulacija za zaplete HA_13

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.377	0.373	0.344
Moč [odvisnost]	0.141	0.089	0.110

1.14) `p_zaplet_HA_14 =  c(rep(c(0.40,0.50),5),rep(0.3,10))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Vsi zapleti se pojavljajo precej enako pogosto, nekoliko pogosteje se pojavlja prvih pet zapletov in sicer rahlo bolj pogosto pri drugi operaciji kot pri prvi. Sicer pa se vsi ostali zapleti pojavljajo pri približno vsakem tretjem bolniku, pri obeh operacijah enako.

Tabela 19: Simulacija za zaplete HA_14

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.531	0.790	0.526
Moč [odvisnost]	0.200	0.229	0.121

1.15) `p_zaplet_HA_15 = c(rep(c(0.45,0.55),10))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Vsi zapleti se pojavljajo precej enako pogosto, nekoliko pogosteje pri drugi operaciji. Vsak izmed zapletov se pojavi pri približno vsakem drugem bolniku, pri obeh operacijah enako.

Tabela 20: Simulacija za zaplete HA_15

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.777	0.999	0.654
Moč [odvisnost]	0.301	0.450	0.094

1.16) `p_zaplet_HA_16 = c(40,50,rep(c(0.40,0.50),4),rep(0.3,10))`

V vzorcu zapletov vidimo rahel vzorec v smislu pogostosti zapletov in njihove povezanosti z operacijo. Vsi zapleti se pojavljajo precej enako pogosto, nekoliko pogosteje se pojavlja prvih pet zapletov in sicer rahlo bolj pogosto pri drugi operaciji kot pri prvi. Sicer pa se vsi ostali zapleti pojavljajo pri približno vsakem tretjem bolniku, pri obeh operacijah enako.

Tabela 21: Simulacija za zaplete HA_16

	Osnovni test	Logistična - seštevanje zapletov	Logistična - večkratno testiranje
Moč	0.485	0.411	0.444
Moč [odvisnost]	0.202	0.116	0.189

Stadiji

Velikosti testov

Najprej si poglejmo kako je z velikostjo ustvarjenih treh testov. Pogledali so bomo velikost testov pod različnimi ničelnimi "domnevami" o stadijih in jih primerjali med seboj.

0.1) `p_stadij_H0_1 = c(rep(0.25,8))`

V vzorcu stadijev ne vidimo vzorca v smislu pogostosti stadijev in njihove povezave z operacijo. Pri vsaki operaciji je vsak četrti bolnik s posameznim stadijem.

Tabela 22: Simulacija za stadije H0_1

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.05	0.049	0.016

0.2) `p_stadij_H0_2 = c(0.13,rep(0.29,3), 0.13, rep(0.29,3))`

V vzorcu stadijev vidimo, da je pri obeh operacijah eden izmed stadijev manj pogost kot ostali trije, ki so približno enako pogosti. Med operacijama ne opazimo razlik.

Tabela 23: Simulacija za stadije H0_2

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.049	0.044	0.013

0.3) `p_stadij_H0_3 = c(0.4,0.4,0.1,0.1,0.4,0.4,0.1,0.1)`

V vzorcu stadijev vidimo, da je pri obeh operacijah eden izmed stadijev manj pogost kot ostali trije, ki so približno enako pogosti. Med operacijama ne opazimo razlik.

Tabela 24: Simulacija za stadije H0_3

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.048	0.046	0.013

Moči testov

Sedaj pa si poglejmo kako je z močjo ustvarjenih treh testov. Pogledali so bomo velikost testov pod različnimi alternativnimi "domnevami" o stadijih in jih primerjali med seboj. Velja opozoriti, da se bomo z verjetnostmi težko "lepo" približali ničelni domnevi, saj zaradi njihove povezanosti ne moremo spreminjati zgolj verjetnosti posameznega stadija.

1.1) `p_stadij_HA_1 = c(0.34,rep(0.22,3),0.40,rep(0.20,3))`

V vzorcu stadijev vidimo, da je pri obeh operacijah eden izmed stadijev nekoliko bolj pogost kot ostali trije, ki so približno enako pogosti. Opazimo tudi, da je pri eni izmed operacij večja razlika v pogostosti stadijev.

Tabela 25: Simulacija za stadije HA_1

	Hi-kvadrat test	Osnovni test	Logistična regresija
Moč	0.128	0.121	0.041

1.2) `p_stadij_HA_2 = c(0.37,rep(0.21,3),0.46,rep(0.18,3))`

V vzorcu stadijev vidimo, da je pri obeh operacijah eden izmed stadijev nekoliko bolj pogost kot ostali trije, ki so bolj redki in enako pogosti. Opazimo tudi, da je pri eni izmed operacij večja razlika v pogostosti stadijev.

Tabela 26: Simulacija za stadije HA_2

	Hi-kvadrat test	Osnovni test	Logistična regresija
Moč	0.23	0.228	0.075

1.3) `p_stadij_HA_3 = c(0.43,rep(0.19,3),0.55,rep(0.15,3))`

V vzorcu stadijev vidimo, da je pri obeh operacijah eden izmed stadijev bolj pogost kot ostali trije, ki so bolj redki in enako pogosti. Opazimo tudi, da je pri eni izmed operacij večja razlika v pogostosti stadijev.

Tabela 27: Simulacija za stadije HA_3

	Hi-kvadrat test	Osnovni test	Logistična regresija
Moč	0.376	0.392	0.156

1.4) `p_stadij_HA_4 = c(0.55,rep(0.15,3),0.7,rep(0.1,3))`

V vzorcu stadijev vidimo, da je pri obeh operacijah eden izmed stadijev precej bolj pogost kot ostali trije, ki so približno enako pogosti in precej redki. Opazimo tudi, da je pri eni izmed operacij večja razlika v pogostosti stadijev.

Tabela 28: Simulacija za stadije HA_4

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.605	0.618	0.213

1.5) `p_stadij_HA_5 <- c(0.67,rep(0.11,3),0.82,rep(0.06,3))`

V vzorcu stadijev vidimo, da je pri obeh operacijah eden izmed stadijev izrazito bolj pogost kot ostali trije, ki so približno enako pogosti in sicer zelo redki. Opazimo tudi, da je pri eni izmed operacij nekoliko večja razlika v pogostosti stadijev.

Tabela 29: Simulacija za stadije HA_5

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.702	0.727	0.292

1.6) `p_stadij_HA_6 <- c(0.55,rep(0.15,3),0.82,rep(0.06,3))`

V vzorcu stadijev vidimo, da je pri obeh operacijah eden izmed stadijev izrazito bolj pogost kot ostali trije, ki so približno enako pogosti in precej redki. Opazimo tudi, da je pri eni izmed operacij opazno večja razlika v pogostosti stadijev.

Tabela 30: Simulacija za stadije HA_6

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.994	0.997	0.875

1.7) `p_stadij_HA_7 = c(0.3,0.3,rep(0.2,2),rep(0.25,4))`

V vzorcu stadijev opazimo, da se pri eni od operacij rahlo pogosteje pojavljata 2 stadija, pri drugi operaciji pa ne vidimo razlik v stadijih.

Tabela 31: Simulacija za stadije HA_7

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.269	0.222	0.093

1.8) `p_stadij_HA_8 = c(0.33,0.33,rep(0.17,2),rep(0.25,4))`

V vzorcu stadijev opazimo, da se pri eni od operacij pogosteje pojavljata 2 stadija, pri drugi operaciji pa ne vidimo razlik v stadijih.

Tabela 32: Simulacija za stadije HA_8

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.66	0.53	0.324

1.9) `p_stadij_HA_9 = c(0.4,0.4,rep(0.1,2),rep(0.25,4))`

V vzorcu stadijev vidimo, da se pri eni od operacij večinoma pojavljata zgolj 2 stadija, pri drugi operaciji pa ne vidimo razlik v stadijih.

Tabela 33: Simulacija za stadije HA_9

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.999	0.992	0.97

1.10) `p_stadij_HA_10 = c(0.5,0.5,rep(0,2),0.6,0.1,0.2,0.1)`

V vzorcu stadijev vidimo, da se pri eni od operacij pojavljata zgolj 2 stadija, pri drugi operaciji pa je občutno najpogostejši eden izmed stadijev (sovpada s prvo operacijo), ostali trije so redki, vendar se pojavijo.

Tabela 34: Simulacija za stadije HA_10

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.274	0.216	0.101

1.11) `p_stadij_HA_11 = c(0.4,0.3,0.2,0.1,rep(0.25,4))`

V vzorcu stadijev vidimo, da se pri eni od operacij v večini pojavljajo trije stadiji, vendar niso enako pogosti, četrti je redek, pri drugi operaciji pa ne opazimo razlik med številom posameznih stadijev.

Tabela 35: Simulacija za stadije HA_11

	Hi-kvadrat test	Osnovni test	Logistična regresija
Velikost	0.948	0.919	0.801

Napotki za preverjanje domnev

Spodaj podani napotki za preverjanje domnev služijo raziskovalcem kot vodilo pri izbiri testa za preverjanje domnev o zapletih in stadijih. Testi oziroma njihova koda v programu R je podana v poglavju Priloge. Podatke v primerni tabelarični obliki, ki je opisana v prilogi, zgolj vstavimo v funkcijo kot vhodni parameter, ta pa nam enostavno sporoči odločitev o ničelni domnevi in poda dodatne informacije.

Preverjanje domnev o zapletih

Iz tabelaričnih prikazov velikosti in moči ustvarjenih testov o zapletih pod različnimi verjetnostmi, tako ničelnimi kot alternativnimi, vidimo, da so velikosti vseh treh obravnavanih testov, ne glede na to ali je pojav posameznega zapleta odvisen od lastnosti bolnika (npr. mlajši bolniki imajo manjšo verjetnost pojava posameznega zapleta) ali pa če posamezen bolnik na verjetnost zapleta ne vpliva, ustrezne tj. manjše od upoštevane stopnje statistične značilnosti $\alpha = 0.05$.

Če pa pogledamo moči testov, iz katerih lahko razberemo kako dober je posamezen test (test z večjo močjo z večjo verjetnostjo zavrne ničelno domnevo v primeru, da podatki izhajajo iz alternativne domneve), hitro vidimo, da so v primeru odvisnosti zapletov od lastnosti bolnika moči precej nižje kot v primeru, ko omenjene odvisnosti ni. Na podlagi tega lahko hitro podamo priporočilo oziroma bolje, opozorilo, da če iz podatkov vzorca bolnikov in njihovih zapletov pri operacijah vidimo, da prihaja do nekakšne povezanosti zapletov z bolnikovimi lastnostmi (v našem postopku simulacij smo bolnike razdelili na 2 dela), predlagani testi ne bodo prav uporabni, saj bomo imeli precejšnje težave z zavračanjem domnev, četudi bodo podatki izhajali iz alternativnih domnev. Seveda pa to ne pomeni, da so testi neuporabni, vseeno lahko poizkusimo in v kolikor bodo podatki močno odstopali od ničelne domneve, bomo še vedno z veliko verjetnostjo lahko le to zavrnili in s tem potrdili alternativno domnevo. Navsezadnje je bolje uporabiti omenjene teste kot pa jih sploh ne, saj drugih na razpolago nimamo.

V kolikor pa zapleti niso odvisni od bolnikovih lastnosti, temveč zgolj od operacije kateri bolnik pripada (pod alternativno domnevo) so moči relativno visoke (moči lahko primerjamo samo med testi, iz pogleda na moč samo določenega testa pa ne moremo povedati ničesar). Tako lahko rečemo, da so testi veliko uporabnejši tj. imajo večjo moč v primeru, ko zapleti niso povezani z bolnikovimi lastnostmi. Tako se bomo v nadaljevanju osredotočili na primerjavo testov na podlagi njihovih simuliranih moči pod alternativnimi domnevami v primeru neodvisno simuliranih zapletov.

Želimo torej podati neko splošno priporočilo o uporabi testov za preverjanje domneve, da število zapletov bolnika ni odvisno od operacije kateri bolnik pripada. Spomnimo se, da imamo tri teste, katere smo poimenovali, 1 - Osnovni test (`test_osnovni_zapleti`), 2 - Logistično seštevanje (`test_logisticna_sum_zapleti`) in 3 - Logistično večkratno testiranje (`test_logisticna_veckratno_zapleti`). Opomniti velja, da testi ne odgovarjajo na povsem enako vprašanje oziroma imajo specifiko pri odgovoru na vprašanje o zavrnitvi domneve, kar velja upoštevati pri izbiri testa za preverjanje domnev. Vsi trije testi nam podajo moč oziroma velikost testa ter operacijo z več zapleti, ločijo pa se glede na obravnavo preštevanja zapletov. Osnovni test in Logistično večkratno testiranje pri operacijah namreč primerjata vsako vrsto zapleta posebej (1, 2, . . . , 10) in domnevo zavrneta v kolikor se vsoti statistično značilnih vrst zapletov med operacijama razlikujeta (vsaka vrsta zapleta je lahko statistično značilno bolj pogosta pri eni izmed operacij - operaciji prištejemo 1, ali pa ni statistično značilne razlike - obe operaciji dobita 0). Tako nam v bistvu ne odgovorita povsem na vprašanje, pri kateri operaciji se pojavi več zapletov (v kolikor se) ampak zgolj pri kateri operaciji se zgodi več vrst zapletov. Podata nam namreč lahko, da se več vrst zapletov pojavi pri operaciji 1, čeprav je zapletov v resnici več pri operaciji 2 (pa morda sploh ne statistično značilno). Na drugi strani pa nam test Logističnega seštevanja poda odgovor, ki vsebuje drugačno informacijo in sicer obravnava vse vrste zapletov kot nekakšen skupni zaplet in pri seštevanju zapletov glede na operacijo ne loči med njihovimi vrstami. Tako nam zgolj ta test resnično odgovori na vprašanje pri kateri operaciji so zapleti statistično značilno bolj pogosti.

Kot rečeno je velikost vseh treh testov ustrezna, torej manjša ali enaka izbrani stopnji značilnosti $\alpha = 0.05$. Posledično je v primeru, ko v podanih podatkih bolnikov (njihova pripadnost operaciji in beleženje zapletov) ne vidimo vzorca v smislu pogostosti posamezne vrste zapleta in njihove povezanosti z operacijo, torej med operacijama ne moremo razločiti glede na podatke o posamezni vrsti zapleta, načeloma vseeno katerega od treh testov za preverjanje domneve izberemo. Vseeno pa se tudi v zgoraj opisanem primeru velja odločiti za test z največjo močjo, saj so razlike v podatkih med ničelno in alternativnimi domnevami lahko tako majhne, da jih iz samega vzorca ne moremo "potrditi", upoštevati pa velja tudi možnost naključja. V nadaljevanju si bomo ogledali, pri kakšni vrsti podatkov velja izbrati kateri test za preverjanje domneve, da vrsta operacije in število zapletov nista povezana. Upoštevali bomo, da zapleti niso povezani z lastnostmi bolnika, čeprav pa tudi ob upoštevanju le tega ne prihaja do večjih odstopanj pri izbiri testa in izbiro testov načeloma lahko posplošimo. Kot je bilo omenjeno tudi zgoraj, pri izbiri testa velja upoštevati tudi njegovo specifičnost odgovora na zastavljeno vprašanje.

- V kolikor v podatkih vidimo, da se v večini pojavlja zgolj eden izmed zapletov (ne glede na njegovo pogostost), za odtenek pogosteje pri eni izmed operacij, ostali zapleti pa se pojavijo zelo redko (iz podatkov težko razberemo pod katero domnevo so "generirani"), se velja odločiti za Osnovni test ali pa za Logistično večkratno testiranje, med njima ni večjih razlik v moči. Logističnega seštevanja se ne priporoča, saj težko loči med ničelno in alternativno domnevo v primeru, ko sta ti dve zelo blizu, medtem ko ostala dva testa tudi v takem primeru precej dobro razločita med domnevama.
- S povečevanjem razlik pogostosti zapleta med operacijama, kjer se ostali zapleti še vedno pojavljajo zelo redko, pa se razlika med testi zmanjšuje. Tako v primeru podatkov, kjer se v veliki večini pojavlja zgolj eden izmed zapletov in je razlika v njegovi pogostosti med operacijama dobro opazna, pa se velja odločiti tudi za Logistično seštevanje, katerega moč ni občutno manjša od ostalih dveh testov. V tem primeru utegne upoštevati tudi kakšen odgovor na vprašanje v resnici želimo in to lahko pretehta za odtenek manjšo moč testa.
- V primeru podatkov, ko se v večini pojavljata dva zapleta, eden pogosteje, drugi redkeje, ostali zapleti pa so zelo redki, velja upoštevati naslednje. Najprej si pogledamo kako pogosta sta zapleta. V kolikor vidimo, da je eden izmed zapletov zelo pogost, drugi pa precej manj in da je eden od zapletov bolj pogost pri eni izmed operacij, drugi pa pri drugi, so moči testov zelo blizu (najmanjša ostaja pri Logističnem seštevanju) in velja pretehtati med vrsto odgovora in močjo testa. V kolikor pa sta oba zapleta pogostejša pri isti operaciji, pa je razlika med močmi večja in je morda, če nismo zelo omejeni glede odgovora, varneje izbrati Osnovni test ali pa Logistično večkratno testiranje. Na drugi strani pa, če v podatkih vidimo, da sta oba zapleta približno enako pogosta, ne glede na njuno splošno pogostost in nagnjenost h operaciji, velja uporabiti Osnovni test ali pa Logistično večkratno testiranje. V tovrstnem primeru je ob predpostavki, da v podatkih ne opazimo prave razlike med operacijama, Logistično seštevanje zapletov neuporaben test, saj je njegova moč izjemno majhna.
- Logistično seštevanje zapletov povečuje svojo relativno moč glede na ostala zapleta s povečevanjem števila pogostih zapletov in ostala dva zapleta preseže, če "vsi" zapleti kažejo h isti operaciji medtem ko ostali zapleti ne ločijo med operacijama. Če torej v podatkih opazimo večje število zapletov, ki je pogostejše pri eni od operacij, pri ostalih zapletih pa med operacijama ne ločimo, je priporočena uporaba Logističnega seštevanja. Brž ko pa so nekateri zapleti bolj pogosti pri eni, drugi pa pri drugi operaciji, pa se strogo priporoča uporabo Osnovnega testa oziroma Logističnega večkratnega testiranja.

Preverjanje domnev o stadijih

Za preverjanje ničelne domneve, da odločitev za vrsto operacije ni bila povezana s stadijem bolnika smo ustvarili tri različne teste. Poimenovali smo jih 1 - Hi-kvadrat test (`test_hi2_stadiji`), 2 - Osnovni test (`test_osnovni_stadiji`) in 3 - Logistično testiranje (`test_logisticna_stadiji`). Tudi v primeru stadijev imajo testi določeno specifiko pri njihovem odgovoru glede na ničelno domnevo. Hi-kvadrat test nam zgolj pove ali ničelno domnevo zavrnemo, torej ali stadij je statistično značilno povezan z vrsto operacije, ali pa ničelno domnevo obdržimo. Na drugi strani pa nam tako Osnovni test kot Logistično testiranje podata odločitev glede ničelne domneve tj. ali stadij je povezan z vrsto operacije ter obenem za vsak stadij povesta h kateri operaciji je nagnjen. Ničelno domnevo zavrnemo, če je vsaj eden izmed stadijev statistično značilno nagnjen h eni od operacij. Tudi tu seveda velja upoštevati specifiko odgovora pri izbiri testa za preverjanje domnev. Opomnimo še, da testiranje domneve o stadijih ni enako testiranju domnev o zapletih, saj velja, da so pri vsakem bolniku stadiji popolnoma linearno odvisni, saj je eden izmed stadijev (recimo četrti) vedno popolna linearna kombinacija ostalih treh (bolnik je lahko samo v enem stadiju). Posledično tudi testi niso povsem podobni tistim s katerimi testiramo domneve o zapletih.

Tako kot pri zapletih, tudi tu želimo neko splošno priporočilo o uporabi testov za preverjanje domneve, da odločitev za vrsto operacije ni bila povezana s stadijem bolnika. Za ustreznost testov mora kot prvo seveda veljati predpostavka o velikosti in sicer, da je verjetnost napake prve vrste, torej verjetnost, da ničelno domnevo zavrnemo v kolikor ji podatki pripadajo, manjša od izbrane stopnje značilnosti $\alpha = 0.05$. Pri vseh treh testih smo iz simulacij pod različnimi ničelnimi domnevami (podatke smo generirali na tri različne načine pod ničelno domnevo) ugotovili, da je predpostavka o velikosti testov izpolnjena. Velikost Hi-kvadrat in Osnovnega testa je zelo blizu 0.05, kar je dober znak za v nadaljevanju preverjano moč testa, medtem ko je velikost Logističnega testiranja precej majhna, kar nas sicer v primeru velikosti ne obremenjuje, vendar pa ne kaže dobrega znaka glede moči testa.

Ugotovili smo torej, da so vsi trije testi primerni za testiranje domneve o zapletih. Sedaj pa si poglejmo njihove moči iz katerih razberemo kateri test je najprimernejši. Opomnimo še enkrat, da favoriziramo test z največjo močjo, saj nam ta zagotavlja pravilno odločitev glede ničelne domneve (zavrnitev) v kolikor za podatke velja alternativna domneva. Tako kot v primeru zapletov smo tudi tu moči testov primerjali z generiranjem podatkov pod različnimi alternativnimi domnevami (11 različnih alternativnih domnev) in opazovali kako se spreminjajo njihove moči in katerega izmed testov favoriziramo pri določeni vrsti podatkov. Opomnimo, da se pri izbiri testa ne velja odločiti zgolj na podlagi njegove moči temveč tudi informacije, ki je podana v njegovem odgovoru glede ničelne domneve.

Na podlagi izvedenih simulacij moči lahko hitro ugotovimo, da je bila skrb o majhni moči Logističnega testiranja, ki se je pojavila pri preverjanju velikosti upravičena. Njegova moč je v prav vseh primerih podatkov generiranih iz alternativnih domnev občutno manjša kot moč ostalih dveh. Pri podatkih generiranih pod verjetnostmi močno stran od ničelne domneve je njegova moč sicer visoka vendar še vedno manjša od moči preostalih dveh testov. Razlog za tako nizko moč (in tudi velikost) je opisan že zgoraj. Zaključimo lahko, da je Logistično testiranje najslabši izmed ustvarjenih testov in njegove uporabe ne priporočamo v nobenem primeru.

Velikosti Hi-kvadrat in Osnovnega testa sta si v vseh primerih precej podobni. Je pa med njima nekaj razlik, ki bi lahko odločale o izbiri testa, opišimo jih spodaj in podajmo predlog o uporabi testa.

- V primeru podatkov, kjer so pri obeh operacijah stadiji precej enako pogosti, vendar s podrobnejšim ogledom opazimo, da je pri obeh operacijah nekoliko bolj pogost eden izmed stadijev, pri obeh isti, ter obenem za odtenek pogostejši pri eni izmed operaciji, velja uporabiti Hi-kvadrat test, saj ima ta za odtenek večjo moč in pri svojem odgovoru poda več informacije.
- V primeru podatkov podobnim tistim v prejšnji alineji, vendar pa sta opisana stadija bolj pogosta in tako ostali stadiji redkejši, velja premisliti o uporabi testa in pretehtati med močjo in informacijo v odgovoru. Velja namreč, da s povečevanjem pogostosti enega izmed stadijev dominacijo v moči prevzema Osnovni test, katerega odgovor pa vsebuje manj informacije glede stadijev in njihove povezanosti z informacijo. Razlika v moči sicer ni velika, vendar pa je odločitev o izbiri testa na podlagi naročnika raziskave in njegove preference o moči testa oziroma informaciji odgovora.
- V vseh ostalih primerih se priporoča uporaba Hi-kvadrat testa, ki ima za odtenek večjo moč in njegov odgovor vsebuje več informacije, ki bi naročnika raziskave utegnila zanimati.

Psevdokoda

Zapleti

Generiranje zapletov

[narediZaplet(p)]

- Funkcija sprejme vnaprej definiran vektor verjetnosti p=(p__1, …, p__20) kjer je p__i v [0,1]
- Ustvarimo vektor, ki nam za vsakega bolnika pove na kateri operaciji je - 150 * 0 (prva operacija) in 150 * 1 (druga operacija).
- Za vsak zaplet i naredimo vektor dolžine 150+150 ki nam za vsakega bolnika pove ali se je določen zaplet pri njem zgodil (1) ali se ni (0). Prvih 150 vrednosti je simulirano iz Ber(p__i) tj. Bin(150,p__i) za i = 2n+1 in drugih 150 vrednosti je simulirano iz Ber(p__i) tj. Bin(150,p__i), za i = 2n.
- Funkcija vrne podatkovni okvir z 11 stolpci (operacija, zapleti) in 300 vrsticami (bolniki).

Generiranje zapletov z odvisnostjo

[narediZapletOdvisnost(p__odvisnost,p)]

- Funkcija sprejme vnaprej definirana vektorja verjetnosti p__odvisnost v [0, inf) in p=(p__1, …, p__20) kjer je p__i v [0,1].
- Za vsakega bolnika, ne glede na operacijo določimo, ali je v skupini s spremenjeno verjetnostjo zapletov (0) ali v skupini z začetno verjetnostjo zapletov (1).
- Prvih 150 bolnikov pripada prvi operaciji (0) in drugih 150 bolnikov drugi operaciji (1).
- Za vsakega bolnika, ne glede na vrsto operacije pogledamo v kateri skupini glede odvisnosti zapletov je in pogojno na skupino za vsak zaplet izmed desetih z ustrezno verjetnostjo (skupina 0 -> začetna verjetnost, skupina 1 -> začetna verjetnost * p__odvisnost) žrebamo ali se je zaplet zgodil ali ne.
- Funkcija vrne podatkovni okvir z 11 stolpci (operacija, zapleti) in 300 vrsticami (bolniki).

Verjetnost napake I. vrste brez popravka

[simulacija__zapleti__H0(p)]

- Funkcija sprejme vnaprej definiran vektor verjetnosti p pod ničelno domnevo, da nobeden od zapletov ni povezan z vrsto operacije.
- Znotraj funkcije kličemo `narediZaplet(p)`.
- Za vsak zaplet i=1, …, 10 s hi-kvadrat testom (brez Yatesovega popravka) testiramo ničelno domnevo, da i-ti zaplet in vrsta operacije nista povezana.
- Funkcija vrne vrednost 1, če je vsaj ena od vrednosti p manjša od 0.05 in vrednost 0, če so vse vrednosti $p > 0.05$.

[simulacija__2__zapleti__H0(p)]

- Nekajkrat (1000×) ponovimo simulacijo `simulacija_zapleti_H0` in vrednosti shranjujemo v nov vektor.
- Verjetnost napake 1. vrste izračunamo kot povprečno število zavrnitev ničelne domneve.

Osnovni test

[simulacija__zapleti__smeri(p)]

- Ustvarimo dva vektorja dolžine 10, ki za vsako od operacij beležita, če je pri njej nek zaplet statistično značilno bolj verjeten.
- Znotraj funkcije kličemo `narediZaplet(p)`.
- Za vsakega od zapletov s hi-kvadrat testom (brez Yatesovega popravka) preverimo ničelno domnevo, da i-ti zaplet in vrsta operacije nista povezana.
- Na podlagi vrednosti p popravljene z Bonferronijevim popravkom se pri vsakem testu odločimo o zavrnitvi ničelne domneve.

- Pri vsakem zapletu zabeležimo pri kateri operaciji je bil bolj pogost (1 = 1. operacija, 2 = 2. operacija, 0 = zaplet in operacija nista statistično značilna povezana).
- Funkcija vrne `TRUE`, če je število zapletov statistično značilno odvisno od operacije (različno število zapletov) in vrsto operacije, pri kateri se zgodi več zapletov (0/1/2).

[simulacija_2_zapleti(p)]

- Ustvarimo vektorja, ki nam beležita informacijo o zavrnitvi domnev glede zapletov in smereh zavrnitev.
- Nekajkrat (1000×) ponovimo `simulacija_zapleti_smeri(p)` in shranjujemo informacije o zavrnitvah in smereh zavrnitev.
- Moč oziroma velikost testa izračunamo kot povprečno število zavrnitev ničelne domneve, da število zapletov ni statistično značilno odvisno od operacije.
- Izpišemo tudi pri kateri operaciji se pojavi več zapletov (če se).

Logistično seštevanje

[simulacija_zapleti_logisticna_sum(p)]

- Znotraj funkcije kličemo `narediZaplet(p)`.
- Seštejemo število zapletov pri vsakem bolniku in jih shranimo v vektor.
- Ustvarimo logistični model, kjer odvisno spremenljivko predstavlja vrsta operacije (1/2) in neodvisno spremenljivko število zapletov za vsakega bolnika.
- S pomočjo vrednosti p pri parametru modela β_1 preverimo ali zavrnemo domnevo o ničelni povezanosti števila zapletov in vrste operacije.
- Vrnemo informacijo o zavrnitvi domneve, da operacija ni statistično značilno povezana s številom zapletov (T/F) in pri kateri operaciji se zgodi več zapletov (0/1/2).

[simulacija_2_zapleti_logisticna_sum(p)]

- Ustvarimo vektorja, ki nam beležita informacijo o zavrnitvi domnev glede zapletov in smereh zavrnitev.
- Nekajkrat (1000 ×) ponovimo `simulacija_zapleti_logisticna_sum(p)` in shranjujemo informacije o zavrnitvah in smereh zavrnitev.
- Moč oziroma velikost testa izračunamo kot povprečno število zavrnitev ničelne domneve, da število zapletov ni statistično značilno odvisno od operacije.
- Izpišemo tudi pri kateri operaciji se pojavi več zapletov (če se).

Logistična večkratno testiranje

[simulacija_zapleti_logisticna(p)]

- Znotraj funkcije kličemo `narediZaplet(p)`.
- Ustvarimo logistični model, kjer odvisno spremenljivko predstavlja vrsta operacije (1/2) in neodvisne spremenljivke zapleti (0/1). Ker testiramo več ničelnih domnev za pravilen izračun vrednosti p uporabimo ukaz glht iz paketa multcomp.
- Iz povzetka modela za vsak zaplet shranimo vrednost p in pripadajoč koeficient.
- Za vsak zaplet preverimo ali je povezan z vrsto operacije ($p < 0.05$) in, si zabeležimo h kateri operaciji je nagnjen (koef > 0 -> 2, koef < 0 -> 1).
- Vrnemo informacijo o zavrnitvi domneve, da operacija ni statistično značilno povezana s številom zapletov (T/F) in pri kateri operaciji se zgodi več zapletov (0/1/2).

[simulacija_2_zapleti_logisticna(p)]

- Ustvarimo vektorja, ki nam beležita informacijo o zavrnitvi domnev glede zapletov in smereh zavrnitev.
- Nekajkrat (1000×) ponovimo `simulacija_zapleti_logisticna(p)` in shranjujemo informacije o zavrnitvah in smereh zavrnitev.
- Moč oziroma velikost testa izračunamo kot povprečno število zavrnitev ničelne domneve, da število zapletov ni statistično značilno odvisno od operacije.
- Izpišemo tudi pri kateri operaciji se pojavi več zapletov (če se).

Stadiji

Generiranje stadijev

[somethingStadij(p)]

- Funkcija sprejme vnaprej definiran vektor verjetnosti p=(p__1, . . . , p__8), kjer je p1+p2+p3+p4=1 in p5+p6+p7+p8=1.
- Sprva ustvarimo vektor stadij (1,2,3,4), dolžine 150+150, kjer je prvih 150 vrednosti vzorčenih iz 1:4 z verjetnostmi p__i za i=1, . . . , 4, s ponavljanjem in drugih 150 vrednosti vzorčenih iz 1:4 z verjetnostmi p__i za i=5, . . . , 8, s ponavljanjem.
- Za vsak stadij ustvarimo vektor dolžine 300 stadij__i, ki vsakemu bolniku priredi vrednost 1, če je bil v i-tem stadiju in vrednost 0, če ni bil v i-tem stadiju.
- Funkcija vrne podatkovni okvir s 5 stolpci (operacija, stadiji) in 300 vrsticami (bolniki).

Verjetnost napake I. vrste brez popravka

[simulacija__stadiji__H0(p)]

- Funkcija sprejme vnaprej definiran vektor verjetnosti p pod ničelno domnevo, da nobeden od stadijev ni povezan z vrsto operacije tj. za vsak stadij se ujema verjetnost prve in druge operacije, ter verjetnosti za stadije za prvo operacijo se seštejejo v 1, enako za drugo operacijo.
- Znotraj funkcije kličemo `narediStadij(p)`.
- Za vsak stadij i=1, . . . , 4 s hi-kvadrat testom (brez Yatesovega popravka) testiramo ničelno domnevo, da i-ti stadij in vrsta operacije nista povezana.
- Funkcija vrne vrednost 1, če je vsaj ena od vrednosti p manjša od 0.05 in vrednost 0, če so vse vrednosti p večje od 0.05.

[simulacija__2__stadiji__H0(p)]

- Nekajkrat (10.000×) ponovimo simulacijo `simulacija_stadiji_H0` in vrednosti shranjujemo v nov vektor.
- Verjetnost napake 1. vrste izračunamo kot povprečno število zavrnitev ničelne domneve.

Hi-kvadrat test

[simulacija__stadiji__hi2(p)]

- Znotraj funkcije kličemo `narediStadij(p)`.
- Za vektor v katerem je za vsakega bolnika podan stadij (1/2/3/4) in vektor v katerem je za vsakega bolnika podana vrsta operacije (0/1) s hi-kvadrat testom preverimo domnevo o povezanosti stadijev z operacijo.
- Funkcija vrne TRUE, če ničelno domnevo zavrnemo in FALSE če ničelno domnevo obdržimo.

[simulacija__2__stadiji__hi2(p)]

- Nekajkrat (1000×) ponovimo `simulacija_stadiji_hi2(p)` in v vektor shranjujemo informacije o odločitvah glede ničelne domneve.
- Moč oziroma velikost testa izračunamo kot povprečno število zavrnitev ničelne domneve, da stadiji niso povezani z vrsto operacije.

Osnovni test

[simulacija__stadiji(p)]

- Ustvarimo dva vektorja dolžine 4, ki za vsako od operacij beležita, če je pri njej nek stadij statistično značilno bolj verjeten.
- Znotraj funkcije kličemo `narediStadij(p)`.
- Za vsakega od stadijev s hi-kvadrat testom (brez Yatesovega popravka) preverimo ničelno domnevo, da i-ti stadij in vrsta operacije nista povezana.

- Na podlagi vrednosti p popravljene z Bonferronijevim popravkom se pri vsakem testu odločimo o zavrnitvi ničelne domneve.
- Pri vsakem stadiju zabeležimo pri kateri operaciji je bil bolj pogost (1 = 1. operacija, 2 = 2. operacija, 0 = zaplet in operacija nista statistično značilna povezana).
- Funkcija vrne vektor dolžine 4, kjer za vsak stadij pove, pri kateri operaciji je bil stadij statistično značilno bolj pogost (0/1/2).

[simulacija_2_stadiji(p)]

- Ustvarimo vektorja, ki nam beležita informacijo o zavrnitvi domnev glede stadijev in smereh zavrnitev posameznega stadija ter podatkovni okvir s štirimi stolpci za beleženje smeri.
- Nekajkrat (1000×) ponovimo `simulacija_stadiji(p)` in v podatkovni okvir vrstično shranjujemo informacije in smereh zavrnitev (0/1/2).
- Moč oziroma velikost testa izračunamo kot povprečno število zavrnitev ničelne domneve, da nobeden od stadijev ni statistično značilno odvisen od operacije (vsota po vrstici je 0).
- Za vsak stadij s t-testom za en vzorec preverimo domnevo, da stadij ni povezan z vrsto operacije: v podatkovnem okviru spremenimo: 1-> -1, 2 -> 1 in dvostransko preverjamo domnevo, da je povprečna vrednost vzorca (stadija po bolnikih) enaka 0.
- Izpišemo velikost oziroma moč testa in za vsak stadij povemo ali je povezan z vrsto operacije in smer povezanosti (1/2).

Logisitčno testiranje

[simulacija_stadiji_logisticna(p)]

- Znotraj funkcije kličemo `narediStadij(p)`
- Ustvarimo logistični model, kjer je operacija odvisna spremenljivka in vektor stadijev (1/2/3/4) neodvisna spremenljivka. Ker je 4. stadij popolna linearna kombinacija prvih treh je potrebno stadije obravnavati kot opisno spremenljivko (faktor).
- Znotraj modela testiramo 4 ničelne domneve, da stadij i=1, . . . ,4 ni povezan z vrsto operacije (modelu dodamo linfct=C).
- Shranimo si vrednosti p in koeficiente modela iz česar razberemo statistično značilnost domnev in operacijo h kateri je stadij nagnjen.
- Funkcija vrne vektor dolžine 4, kjer za vsak stadij pove, pri kateri operaciji je bil stadij statistično značilno bolj pogost (0/1/2).

[simulacija_2_stadiji_logisticna(p)]

- Ustvarimo vektorja, ki nam beležita informacijo o zavrnitvi domnev glede stadijev in smereh zavrnitev posameznega stadija ter podatkovni okvir s štirimi stolpci za beleženje smeri.
- Nekajkrat (1000×) ponovimo `simulacija_stadiji_logisticna(p)` in v podatkovni okvir vrstično shranjujemo informacije in smereh zavrnitev (0/1/2).
- Moč oziroma velikost testa izračunamo kot povprečno število zavrnitev ničelne domneve, da nobeden od stadijev ni statistično značilno odvisen od operacije (vsota po vrstici je 0).
- Za vsak stadij s t-testom za en vzorec preverimo domnevo, da stadij ni povezan z vrsto operacije: v podatkovnem okviru spremenimo: 1-> -1, 2 -> 1 in dvostransko preverjamo domnevo, da je povprečna vrednost vzorca (stadija po bolnikih) enaka 0.
- Izpišemo velikost oziroma moč testa in za vsak stadij povemo ali je povezan z vrsto operacije in smer povezanosti (1/2).

Priloge

Zapleti

Vsi spodnji testi kot vhodni parameter sprejmejo podatkovni okvir, kjer je vsakemu bolniku dodeljena svoja vrstica, stolpci pa vsebujejo informacije glede vrste operacije, na kateri je bolnik (0: 1. operacija, 1: 2. operacija), ter indikator dogodka, da se je posamezen zaplet zgodil. Stolpci so poimenovani kot “operacija”, “zaplet1”, “zaplet2”, . . . , “zaplet10”.

Osnovni test

```
test_osnovni_zapleti <- function(podatki){
  st_zapletov_1 = rep(0, 10)
  st_zapletov_2 = rep(0, 10)
  test1 <- chisq.test(podatki$operacija, podatki$zaplet1, correct = F)
  test2 <- chisq.test(podatki$operacija, podatki$zaplet2, correct = F)
  test3 <- chisq.test(podatki$operacija, podatki$zaplet3, correct = F)
  test4 <- chisq.test(podatki$operacija, podatki$zaplet4, correct = F)
  test5 <- chisq.test(podatki$operacija, podatki$zaplet5, correct = F)
  test6 <- chisq.test(podatki$operacija, podatki$zaplet6, correct = F)
  test7 <- chisq.test(podatki$operacija, podatki$zaplet7, correct = F)
  test8 <- chisq.test(podatki$operacija, podatki$zaplet8, correct = F)
  test9 <- chisq.test(podatki$operacija, podatki$zaplet9, correct = F)
  test10 <- chisq.test(podatki$operacija, podatki$zaplet10, correct = F)
  testi <- list(test1, test2, test3, test4, test5, test6, test7, test8, test9, test10)

  for (i in 1:10){
    if (testi[[i]]$p.value < 0.05/10) {
      if (testi[[i]]$observed[3] > testi[[i]]$expected[3]) {
        st_zapletov_1[i] = 1
      } else {
        st_zapletov_2[i] = 1
      }
    }
  }

  if (sum(st_zapletov_1) > sum(st_zapletov_2)) {
    smer = 1
  } else if (sum(st_zapletov_1) < sum(st_zapletov_2)) {
    smer = 2
  } else {
    smer = 0
  }

  zavrnemo <- sum(st_zapletov_1) != sum(st_zapletov_2)
  if (zavrnemo == T) {
    blabla <- paste("Zapleti so povezani z vrsto operacije in pri operaciji",smer,
                    "se zgodi več vrst zapletov.")
  } else {
    blabla <- "Zapleti niso povezani z vrsto operacije, torej je število vrst zapletov,
    ki se pogosteje pojavijo pri eni izmed operacij, med operacijama enako."
  }
  return(blabla)
}
```

Logistično testiranje

```
test_logisticna_sum_zapleti <- function(podatki) {
  podatki$stZapletov <- apply(podatki[,2:11], 1, sum)

  model <- glht(glm(operacija~stZapletov, data = podatki, family = "binomial"))
  povzetek <- summary(model)

  smer <- 0

  if (povzetek$test$pvalues[2] < 0.05) {
    if (model$coef[2] > 0) {
      smer <- 2
    } else {
      smer <- 1
    }
  }

  zavrnemo = povzetek$test$pvalues[2] < 0.05

  if (zavrnemo == T) {
    blabla <- paste("Zapleti so povezani z vrsto operacije in pri operaciji",smer,
                    "se zgodi več zapletov.")
  } else {
    blabla <- "Zapleti niso povezani z vrsto operacije, torej je število zapletov pri
    operacijah enako."
  }

  return(blabla)
}
```

Logistično večkratno testiranje

```
test_logisticna_veckratno_zapleti <- function(podatki) {
  model <- glht(glm(operacija~., data = podatki, family = "binomial"))
  povzetek <- summary(model)
  p_vrednosti <- povzetek$test$pvalues
  koeficienti <- model$coef

  zavrnitve <- rep(FALSE, 10)
  smeri <- rep(0,10)

  for (i in 2:11) {
    zavrnitve[i] <- (p_vrednosti[i] < 0.05)

    if (zavrnitve[i]) {
      if (koeficienti[i] > 0) {
        smeri[i] = 2
      } else {
        smeri[i] = 1
      }
    }
  }

  smer <- 0

  if (sum(smeri==2) > sum(smeri==1)) {
    smer = 2
  } else if (sum(smeri==2) < sum(smeri==1)) {
    smer = 1
  }

  zavrnemo = sum(smeri==1) != sum(smeri==2)

  if (zavrnemo == T) {
    blabla <- paste("Zapleti so povezani z vrsto operacije in pri operaciji",smer,
                    "se zgodi več vrst zapletov.")
  } else {
    blabla <- "Zapleti niso povezani z vrsto operacije, torej je število vrst zapletov,
    ki se pogosteje pojavijo pri eni izmed operacij, med operacijama enako."
  }

  return(blabla)
}
```

Stadiji

Vsi spodnji testi kot vhodni parameter sprejmejo podatkovni okvir, kjer je vsakemu bolniku dodeljena svoja vrstica, stolpci pa vsebujejo informacije glede vrste operacije, na kateri je bolnik (0: 1. operacija, 1: 2. operacija), ter stadija, v katerem je bolnik (od 1 do 4). Stolpca sta poimenovana kot "operacija" in "stadij".

Hi-kvadrat test

```
test_hi2_stadiji <- function(podatki){

  test <- chisq.test(podatki$operacija, podatki$stadij)

  if (test$p.value < 0.05) {
    vrni <- "Stadij je povezan z vrsto operacije."
  } else {
    vrni <- "Stadij ni povezan z vrsto operacije."
  }

  return(vrni)
}
```

Osnovni test

```
test_osnovni_stadiji <- function(podatki){

  stadij1 <- as.numeric(podatki$stadij==1)
  stadij2 <- as.numeric(podatki$stadij==2)
  stadij3 <- as.numeric(podatki$stadij==3)
  stadij4 <- as.numeric(podatki$stadij==4)
  tmp <- data.frame(podatki$operacija, stadij1, stadij2, stadij3, stadij4)
  st_operacij_1 = rep(0,4)
  st_operacij_2 = rep(0,4)

  test1 <- chisq.test(podatki$operacija, podatki$stadij1, correct = F)
  test2 <- chisq.test(podatki$operacija, podatki$stadij2, correct = F)
  test3 <- chisq.test(podatki$operacija, podatki$stadij3, correct = F)
  test4 <- chisq.test(podatki$operacija, podatki$stadij4, correct = F)
  testi <- list(test1, test2, test3, test4)

  for (i in 1:4){
    if (testi[[i]]$p.value < 0.05/4) {
      if (testi[[i]]$observed[3] > testi[[i]]$expected[3]) {
        st_operacij_1[i] = 1
      } else {
        st_operacij_2[i] = 1
      }
    }
  }
  smer = rep(0,4)
  for (i in 1:4){
    if (st_operacij_1[i] > st_operacij_2[i]) {
      smer[i] = 1
    } else if (st_operacij_1[i] < st_operacij_2[i]) {
      smer[i] = 2
    }
  }
  hehe <- list()
  zavrnemo <- (sum(smer)>0)

  if (zavrnemo) {
    haha <- "Stadij je povezan z vrsto operacije"
  } else {
    haha <- "Stadij ni povezan z vrsto operacije"
  }

  for (i in 1:4){
    if (testi[[i]]$p.value > 0.05 | is.na(testi[[i]]$p.value)) {
      hehe <- c(hehe, paste("in stadij", i, "ne vpliva na vrsto operacije."))
    } else {
      hehe <- c(hehe, paste("in bolniki s stadijem", i, "so nagnjeni k operaciji.",
                            smer[i]))
    }
  }
  return(paste(haha,hehe))
}
```

Logistično testiranje

```
test_logisticna_stadiji <- function(podatki){
  podatki$stadij <- as.factor(podatki$stadij)

  C <- rbind(c(1,0,0,0), c(1,1,0,0), c(1,0,1,0), c(1,0,0,1))
  colnames(C) <- c("beta0", "beta1", "beta2","beta3")
  rownames(C) <- c("mu_1=0","mu_2=0","mu_3=0","mu_4=0")


  model <- glm(operacija~stadij, data = podatki, family = "binomial")
  model_primerjava <- summary(glht(model, linfct = C))

  p_vrednosti <- model_primerjava$test$pvalues
  koeficienti <- model_primerjava$test$coefficients

  zavrnitve <- rep(FALSE, 4)
  smeri <- rep(0,4)

  for (i in 1:4) {
    zavrnitve[i] <- p_vrednosti[i] < 0.05

    if (zavrnitve[i]) {
      if (koeficienti[i] > 0) {
        smeri[i] = 2
      } else {
        smeri[i] = 1
      }
    }
  }

  hehe <- list()

  zavrnemo <- (sum(smeri)>0)

  if (zavrnemo) {
    haha <- "Stadij je povezan z vrsto operacije"
  } else {
    haha <- "Stadij ni povezan z vrsto operacije"
  }

  for (i in 1:4){
    if (smeri[i] == 0) {
      hehe <- c(hehe, paste("in stadij", i, "ne vpliva na vrsto operacije."))
    } else {
      hehe <- c(hehe, paste("in bolniki s stadijem", i, "so nagnjeni k operaciji.",
                            smeri[i]))
    }
  }

  return(paste(haha, hehe))
}
```