

LONDON  
SCHOOL of  
HYGIENE  
& TROPICAL  
MEDICINE



LSHTM Research Online

Bielicki, JA; (2021) Estimating coverage of empiric treatment regimens for childhood bloodstream infection based on routine microbiological data. PhD (research paper style) thesis, London School of Hygiene & Tropical Medicine. DOI: <https://doi.org/10.17037/PUBS.04659846>

Downloaded from: <https://researchonline.lshtm.ac.uk/id/eprint/4659846/>

DOI: <https://doi.org/10.17037/PUBS.04659846>

**Usage Guidelines:**

Please refer to usage guidelines at <https://researchonline.lshtm.ac.uk/policies.html> or alternatively contact [researchonline@lshtm.ac.uk](mailto:researchonline@lshtm.ac.uk).

Available under license. To note, 3rd party material is not necessarily covered under this license: <http://creativecommons.org/licenses/by-nc-nd/3.0/>

<https://researchonline.lshtm.ac.uk>

## Appendix B.

Stata\_DTAmodel\_WISCA.do - Printed on 04.12.2020 13:50:10

```
1
2 ***** STATA program to calculate coverage for a single regimen
3 ***** for a variable number of bacteria
4 ***** 21 October 2016
5
6
7
8 ***** PART 1: Create parameter dataset
9 *****
10
11
12 clear
13
14 ***** Create empty dataset
15
16 set obs 300
17 egen num_row = seq()
18
19
20 ***** Define parameter dimensions (number of rows)
21
22 scalar max_regimen = 1
23 scalar max_bacteria = 10
24
25 label define name_regimen 0 "Incidence" 1 "Amox/genta"
26
27 label define name_bacteria 1 "S.aureus" 2 "S.pneumoniae" 3 "E.faecalis" 4 "E.faecium" 5 "E.coli" 6
   "Enterobacter" 7 "K.pneumoniae" 8 "P.aeruginosa" ///
   9 "Other: high susc" 10 "Other: var susc"
28
29
30 gen max_row = (1+max_regimen) * max_bacteria
31 drop if num_row>max_row
32
33 capture drop name*
34 gen regimen = int((num_row-1)/max_bacteria)
35 gen bacteria = mod(num_row-1,max_bacteria)+1
36
37 label values regimen name_regimen
38 label values bacteria name_bacteria
39
40
41 ***** Define parameters
42
43
44 **** default bacterial incidence
45
46 ** default prior
47 gen incid_priorA = 1
48
49 gen incid_num = .
50 edit bacteria incid_num if regimen==0
51
52 replace incid_num = 449 in 1
53 replace incid_num = 163 in 2
54 replace incid_num = 175 in 3
55 replace incid_num = 95 in 4
56 replace incid_num = 380 in 5
57 replace incid_num = 122 in 6
58 replace incid_num = 183 in 7
59 replace incid_num = 137 in 8
60 replace incid_num = 175 in 9
61 replace incid_num = 57 in 10
62
63
64 egen incid_total = sum(incid_num) if regimen==0
65 gen incid_prop = incid_num / incid_total
66
67 gen incid_postA = incid_priorA + incid_num
```

```
68
69
70 **** default regimen / bacteria susceptibility
71
72 ** default prior
73 gen suscep_priorA = 0.5
74 gen suscep_priorB = 0.5
75
76 *** change suscep_priorA and priorB for any bacteria with intrinsic resistance fo regimen,
specify prior as a beta(1, 9999)
77 /*
78 - replace suscep_priorA = 1 in XX
79 - replace suscep_priorB = 9999 in XX
80 */
81
82 gen suscep_tested = .
83 gen suscep_prop = .
84
85 edit bacteria suscep_tested suscep_prop if regimen==1
86
87 replace suscep_tested = 421 in 11
88 replace suscep_tested = 152 in 12
89 replace suscep_tested = 160 in 13
90 replace suscep_tested = 86 in 14
91 replace suscep_tested = 378 in 15
92 replace suscep_tested = 121 in 16
93 replace suscep_tested = 183 in 17
94 replace suscep_tested = 137 in 18
95 replace suscep_tested = 164 in 19
96 replace suscep_tested = 56 in 20
97 replace suscep_prop = 0.97 in 11
98 replace suscep_prop = 0.97 in 12
99 replace suscep_prop = 0.93 in 13
100 replace suscep_prop = 0.15 in 14
101 replace suscep_prop = 0.88 in 15
102 replace suscep_prop = 0.90 in 16
103 replace suscep_prop = 0.75 in 17
104 replace suscep_prop = 0.76 in 18
105 replace suscep_prop = 0.96 in 19
106 replace suscep_prop = 0.98 in 20
107
108
109 gen suscep_true = round(suscep_tested * suscep_prop,1)
110 gen suscep_postA = suscep_priorA + suscep_true
111 gen suscep_postB = suscep_priorB + (suscep_tested - suscep_true)
112
113
114
115 *****
116
117 foreach var in incid_numb incid_total incid_prop incid_postA {
118     capture drop aa
119     gen aa=.
120     replace aa = `var'[_n-10]
121     replace `var'=aa if `var'==.
122 }
123 *
124
125 save "coverage_param"
126
127
128 *****
129 ***** PART 2 : Calculate coverage for regimen
130 *****
131
132
133 capture program drop coverage_calc
134
```

```
135
136   program define coverage_calc, rclass
137
138     clear
139     use "coverage_param"
140     drop if regimen ==0
141
142     gen incid_rand = uniform()
143     gen susc_rand = uniform()
144
145     gen sim_bugnum = 0
146     replace sim_bugnum = invgammmap(incid_postA,incid_rand)
147
148     bysort regimen: egen sim_bugtotal = sum(sim_bugnum)
149
150     gen sim_incid = sim_bugnum / sim_bugtotal
151
152     gen sim_susc = 0
153     replace sim_susc = invibeta(suscep_postA,suscep_postB, susc_rand)
154
155     gen sim_bugcover = sim_incid * sim_susc
156
157     bysort regimen: egen coverage = sum(sim_bugcover)
158
159     return scalar base=coverage[1]
160
161 end
162
163 *****
164
165 clear
166
167 scalar drop _all
168
169 set seed 312456
170
171 simulate base=r(base) , ///
172     reps(1000) saving(Regimen_1, replace) : coverage_calc
173
174
175 qui summ base
176 disp as text "Coverage = " r(mean)
177
178 qui centile base, centile(2.5 97.5)
179 disp "95% credible interval = " r(c_1) " to " r(c_2)
180
181 *****
182
183
```