Medical-Crisis Checklists in the Emergency Department: A Simulation-Based Multi-Institutional Randomized Controlled Trial Appendix 2: Reproducible Data

Table of Contents

Dryver E, *et al. BMJ Qual Saf* 2021;0:1–9. doi: 10.1136/bmjqs-2020-012740

Data

The data collected for this study and the programs that analyzed these data are publicly available. The data for this study are posted at
https://www.dropbox.com/sh/u1xfkz2s7fyjxsc/AAD3l3ZL1mHqoeRkNDkTCKr0a?dl=0
in an Excel spreadsheet named *Checklists_ED_interventions.xlsx*. It is freely available to anyone. To make a copy of this file, go to this URL, pull down the *Open* box next to *Checklists_ED_interventions.xlsx*, and select "Open in Excel".

Figure 2

Figure 2 was generated by the Stata program *Figure1.do*. It creates graphs called *fifteen*, *boxplot* and *legend*. PowerPoint was used to overlay *fifteen* on top of *boxplot* and to annotate the figure. *Figure1.do* is given in the following monospaced font.

```
log using Figure1.log, replace
* Figure1.log
version 16

import excel  Checklists_ED_interventions.xlsx , sheet("Results") firstrow

* n15_interventions = number of interventions by a team in a particular
* scenario in the first 15 minutes.

gen n15_interventions = 0
foreach intervention of varlist m?_done_15 m10_done_15 {
    replace n15_interventions = n15_interventions + `intervention' ///
        if  `intervention' != .
}

* The variable m_max indicates the total number of
* possible interventions for each scenario at each ED.

list checklist n15_interventions m?_done_15 m10_done_15 m_max in 1/10, table
list checklist n15_interventions m?_done_15 m10_done_15 m_max ///
    if m_max < n15_interventions, table

* Draw preliminary scatter plot of number of interventions by checklist use

scatter n15_interventions checklist
more
codebook scenario

* Summarize interventions by scenarios

forvalues i = 1/8 {
    dis ""
    dis "scenario `i'"
    preserve
    keep if scenario == `i'
    sum m?_done_15 m10_done_15
    restore
}

* percent_int15 = the percentage of interventions performed by each team
* with each scenario in 15 minutes

gen percent_int15 = 100*n15_interventions/m_max

* Draw preliminary scatter plot of % of interventions performed by checklist use

scatter percent_int15 checklist

* Generate x-axis  for scatter plots by scenarios.
* The following local macros will be used to tune the graph.

local check_width = 1
local scenario_width = 1.5
local jiggle = 0.25
gen x = .
forvalues i = 1/8 {
    replace x = (scenario-1)*(`check_width' + `scenario_width') ///
        + checklist*`check_width' if scenario == `i'

* generate veritcal separation lines
```

```
    local line`i' = `scenario_width'/2 + (`i'-1)*(`check_width' + `scenario_width') ///
        + `check_width'
    di " line`i' = " `line`i''
}

* Jiggle % interventions in 15 minutes to avoid collisions in scatter plot

sort scenario checklist percent_int15
by scenario checklist percent_int15, sort : egen float ties15 = count(percent_int15)
by scenario checklist percent_int15: gen j15 = _n
gen jiggle_x15 = x
replace jiggle_x15 = x + (-int(ties15/2) + j15-1)*`jiggle'
replace jiggle_x15 = jiggle_x15 + `jiggle'/2 if mod(ties15,2) == 0


if team <= 10 &              ed != 1 {
    display "team ED conflict, team = " team " ED = " ed
}
if team > 10 &  team <= 20 & ed != 2 {
    display "team ED conflict, team = " team " ED = " ed
}
if team > 20 &  team <= 30 & ed != 3 {
    display "team ED conflict, team = " team " ED = " ed
}
if team > 30 &  team <= 41 & ed != 4 {
    display "team ED conflict, team = " team " ED = " ed
}
scatter percent_int15 x, name(ed1, replace) ylabel(0(10)100, angle(zero))
more
*local maxx = `scenario_width'/2 + (7)*(`check_width' + `scenario_width') + `check_width' + .5
local maxx = `line8' - 0.25
 di "maxx = `maxx'"
 di "line1 = " `line1'
twoway                                                                                  ///
    (scatter percent_int15 jiggle_x15 if team ==1,  mcolor(red)   msymbol(circle))      ///
    (scatter percent_int15 jiggle_x15 if team ==2,  mcolor(blue)  msymbol(circle))      ///
    (scatter percent_int15 jiggle_x15 if team ==3,  mcolor(green) msymbol(circle))      ///
    (scatter percent_int15 jiggle_x15 if team ==4,  mcolor(cyan)  msymbol(circle))      ///
    (scatter percent_int15 jiggle_x15 if team ==5,  mcolor(gold)  msymbol(circle))      ///
    (scatter percent_int15 jiggle_x15 if team ==6,  mcolor(red)   msymbol(circle_hollow))    ///
    (scatter percent_int15 jiggle_x15 if team ==7,  mcolor(blue)  msymbol(circle_hollow))    ///
    (scatter percent_int15 jiggle_x15 if team ==8,  mcolor(green) msymbol(circle_hollow))    ///
    (scatter percent_int15 jiggle_x15 if team ==9,  mcolor(cyan)  msymbol(circle_hollow))    ///
    (scatter percent_int15 jiggle_x15 if team ==10, mcolor(gold)  msymbol(circle_hollow))    ///
                                                                                        ///
    (scatter percent_int15 jiggle_x15 if team ==11,  mcolor(red)   msymbol(square))     ///
    (scatter percent_int15 jiggle_x15 if team ==12,  mcolor(blue)  msymbol(square))     ///
    (scatter percent_int15 jiggle_x15 if team ==13,  mcolor(green) msymbol(square))     ///
    (scatter percent_int15 jiggle_x15 if team ==14,  mcolor(cyan)  msymbol(square))     ///
    (scatter percent_int15 jiggle_x15 if team ==15,  mcolor(gold)  msymbol(square))     ///
    (scatter percent_int15 jiggle_x15 if team ==16,  mcolor(red)   msymbol(square_hollow))   ///
    (scatter percent_int15 jiggle_x15 if team ==17,  mcolor(blue)  msymbol(square_hollow))   ///
    (scatter percent_int15 jiggle_x15 if team ==18,  mcolor(green) msymbol(square_hollow))   ///
    (scatter percent_int15 jiggle_x15 if team ==19,  mcolor(cyan)  msymbol(square_hollow))   ///
    (scatter percent_int15 jiggle_x15 if team ==20,  mcolor(gold)  msymbol(square_hollow))   ///
                                                                                        ///
    (scatter percent_int15 jiggle_x15 if team ==21,  mcolor(red)   msymbol(triangle))   ///
    (scatter percent_int15 jiggle_x15 if team ==22,  mcolor(blue)  msymbol(triangle))   ///
    (scatter percent_int15 jiggle_x15 if team ==23,  mcolor(green) msymbol(triangle))   ///
    (scatter percent_int15 jiggle_x15 if team ==24,  mcolor(cyan)  msymbol(triangle))   ///
    (scatter percent_int15 jiggle_x15 if team ==25,  mcolor(gold)  msymbol(triangle))   ///
    (scatter percent_int15 jiggle_x15 if team ==26,  mcolor(red)   msymbol(triangle_hollow)) ///
    (scatter percent_int15 jiggle_x15 if team ==27,  mcolor(blue)  msymbol(triangle_hollow)) ///
    (scatter percent_int15 jiggle_x15 if team ==28,  mcolor(green) msymbol(triangle_hollow)) ///
    (scatter percent_int15 jiggle_x15 if team ==29,  mcolor(cyan)  msymbol(triangle_hollow)) ///
    (scatter percent_int15 jiggle_x15 if team ==30,  mcolor(gold)  msymbol(triangle_hollow)) ///
                                                                                        ///
    (scatter percent_int15 jiggle_x15 if team ==31,  mcolor(red)   msymbol(diamond))    ///
    (scatter percent_int15 jiggle_x15 if team ==32,  mcolor(blue)  msymbol(diamond))    ///
    (scatter percent_int15 jiggle_x15 if team ==33,  mcolor(green) msymbol(diamond))    ///
    (scatter percent_int15 jiggle_x15 if team ==34,  mcolor(cyan)  msymbol(diamond))    ///
    (scatter percent_int15 jiggle_x15 if team ==35,  mcolor(gold)  msymbol(diamond))    ///
    (scatter percent_int15 jiggle_x15 if team ==36,  mcolor(red)   msymbol(diamond_hollow))  ///
    (scatter percent_int15 jiggle_x15 if team ==37,  mcolor(blue)  msymbol(diamond_hollow))  ///
    (scatter percent_int15 jiggle_x15 if team ==38,  mcolor(green) msymbol(diamond_hollow))  ///
    (scatter percent_int15 jiggle_x15 if team ==39,  mcolor(cyan)  msymbol(diamond_hollow))  ///
    (scatter percent_int15 jiggle_x15 if team ==40,  mcolor(gold)  msymbol(diamond_hollow))  ///
```

```
    (scatter percent_int15 jiggle_x15 if team ==41,  mcolor(gray)  msymbol(diamond))         ///
    ,  ytitle(% of Interventions In 15 Minutes) ylabel(0(10)100, angle(zero))                ///
       name(fifteen)  xtitle("") legend(off) xlabel(none) xsize(7.5in) aspectratio(0.356)    ///
       xline(`line1' `line2' `line3' `line4' `line5' `line6' `line7' `line8', lcolor(black)) ///
       graphregion(fcolor(white) lcolor(white)) xscale(range(0 `maxx'))
save Checklists_ED_interventions.dta, replace

* Generate legend symbols to be copied and edited in PowerPoint
gen percent_int15_check = percent_int15
replace percent_int15_check = . if checklist == 0
gen percent_int15_nocheck = percent_int15
replace percent_int15_nocheck = . if checklist == 1
graph box percent_int15_nocheck percent_int15_check,                                      ///
    over(scenario) box(1, color(black) fcolor(none)) box(2, color(black) fcolor(none))  ///
    ytitle(% of Interventions In 15 Minutes) ylabel(0(10)100, angle(zero))               ///
    name(boxplot) legend(off) xsize(7.5in) aspectratio(0.356)                            ///
    graphregion(fcolor(white) lcolor(white))

clear
set obs 8
gen x = 1
replace x = 1.25 if _n > 4
gen ED = _n
replace ED = _n -4 if _n > 4
twoway (scatter ED x if ED==4 & x ==1, mcolor(black) msymbol(circle))             ///
       (scatter ED x if ED==4 & x ==1.25, mcolor(black) msymbol(circle_hollow))   ///
       (scatter ED x if ED==3 & x ==1, mcolor(black) msymbol(square))             ///
       (scatter ED x if ED==3 & x ==1.25, mcolor(black) msymbol(square_hollow))   ///
       (scatter ED x if ED==2 & x ==1, mcolor(black) msymbol(triangle))           ///
       (scatter ED x if ED==2 & x ==1.25, mcolor(black) msymbol(triangle_hollow)) ///
       (scatter ED x if ED==1 & x ==1, mcolor(black) msymbol(diamond))            ///
       (scatter ED x if ED==1 & x ==1.25, mcolor(black) msymbol(diamond_hollow))  ///
     , xscale(range(0 7.5))  yscale(range(0 5)) name(legend)

clear
log close
```

Figure 3

This figure was created by the program *Figure2.do*. It uses the data file *Checklists_ED_interventions.dta*, which is created by *Figure1.do*. Your must run *Figure1.do* before running *Figure2.do*. *Figure2.do* is given in the following monospaced font.

```
program define medianInterventionsPerformed, rclass
    args times n_teams

* Calculate the median % of interventions performed in teams that
* did, or did not, use the checklist.

* This program was written to enable the calculation of bootstrapped confidence intervals

* Determine a list of all of the unique times at which interventions occured in the first 15
minutes.
* Set the first and last time to be 0 and 15 minutes, respectively.  These times
* will be used in all bootstrapped samples

* This program requires that the data be in wide format.
* A temporary file with the macro name `times' must
* already exist that contains the unique times that interventions were performed in the real
data
* prior to 15 minutes. Note that in the bootstrapped samples there will be times in this file
when no
* intervention was made.

version 16
preserve
* tabulate checklist
forvalues i = 1/`n_teams' {
    local check`i' = checklist[`i']
    local scen`i'  = scenario[`i']
}
* Reformat the data so that there is one record for each intervention
* The team id must be unique. For boostrapped samples we will redefine
* this id to ensure that this is true.
quietly gen id_boot = _n
quietly reshape long m_sec, i(id_boot) j(intervention)
sort id_boot m_sec
quietly drop if m_sec == .

* convert time to minutes
quietly gen m_min = m_sec / 60
label variable m_min "Time since start of scenario (min)"

* Drop data beyond 15 minutes
quietly drop if m_min > 15

* count is the number of interventions performed by each team by time m_ses
by id_boot: gen count = _n

* Expand time so that everyone starts with 0 interventions and ends with
* their maximum number of interventions by 15 minutes

local littleN = _N
local bigN = _N + `n_teams'*2
quietly set obs `bigN'
forvalues i = 1/`n_teams' {
    quietly replace id_boot = `i'           if _n == `littleN' + `i'
    quietly replace m_min = 0               if _n == `littleN' + `i'
    quietly replace count = 0               if _n == `littleN' + `i'
    quietly replace checklist = `check`i''  if _n == `littleN' + `i'
    quietly replace scenario = `scen`i''    if _n == `littleN' + `i'
    quietly sum count if id_boot == `i'
    local maxcount`i' = r(max)
}
forvalues i = 1/`n_teams' {
    quietly replace id_boot = `i'       if _n == `littleN' + `n_teams' + `i'
    quietly replace m_min = 15    if _n == `littleN' + `n_teams' + `i'
    quietly replace count = `maxcount`i''   if _n == `littleN' + `n_teams' + `i'
    quietly replace checklist = `check`i''  if _n == `littleN' + `n_teams' + `i'
    quietly replace scenario = `scen`i''    if _n == `littleN' + `n_teams' + `i'
}
sort id_boot m_min
* m_max is the maximum number of possible interventions for the current id.
```

```
* It is a function of the scenario and the ED.
* There is an intervention used in some scenarios that was not available in one ED.

quietly replace m_max = m_max[_n+1] if m_min == 0
quietly replace m_max = m_max[_n-1] if m_min == 15
* percent is the percent of interventions prformed by each team by time m_min
* on the assigned senario

quietly gen percent = 100*count/m_max
label variable percent "% of interventions performed"
tempfile long
quietly save "`long'", replace
clear
use "`times'"
quietly sum m_min, detail
quietly expand `n_teams'
sort m_min
quietly gen  id_boot = .
quietly replace id_boot = 1 if m_min != m_min[_n-1]
quietly replace id_boot = id_boot[_n-1] +1 if id_boot == .
quietly merge 1:m m_min id_boot using "`long'"
sort  id_boot checklist
quietly replace checklist = checklist[_n-1] if checklist == .
sort checklist id_boot m_min percent
quietly drop if m_min == m_min[_n+1]

quietly replace percent = percent[_n-1] if percent ==.
collapse (median) percent , by(checklist m_min)
sort checklist m_min percent
forvalues i =1/$num_times {
        return scalar timeNoCheck`i' = percent[`i']
        local j = $num_times + `i'
        return scalar timeWithCheck`i' = percent[`j']
}
end //------------------------------------------------

*################# Start of program ###########################################################

log using Figure2.log, replace
* Figure2.log

* Graph the average % of interventions performed for each scenario by teams that
* are, and are not, using checklists as a function of time.
* Stop graph at 15 minutes

* n_teams is the number of ED teams participating in this trial

* This program uses percentile-based confidence intervals.

 local n_teams = 76

set seed 6616157
use Checklists_ED_interventions.dta
*N.B. Checklists_ED_interventions.dta was created by Figure1.do, which must be run before
*     running Figure2.do.

* Check if intervention time is given whenever the intervention is coded as being performed
* or if intervention time is given when the intervention is not performed

forvalues i = 1/10 {

* List if intervention is not done by time of intervention is given
    list id ed scenario checklist m`i'_done m`i'_sec if m`i'_done == 0 & m`i'_sec != .
* List if intervention is done by time of intervention is not given
    list id ed scenario checklist m`i'_done m`i'_sec if m`i'_done == 1 & m`i'_sec == .
}
sort id

* Calculate time to intervention from Canon and Sony times in minutes
* Note that Stata times are in miliseconds

forvalues i = 1/10 {
    gen m`i'_sec_canon = ( m`i'_time_canon- start_time_canon)/1000
    gen m`i'_sec_sony = (m`i'_time_sony- start_time_sony)/1000
* List if times are given for both timers
    list id  m`i'_sec_canon  m`i'_sec_sony if m`i'_sec_canon !=. &  m`i'_sec_sony != .
* List if no time is recorded but intervention was performed
```

```
    list id  m`i'_sec_canon  m`i'_sec_sony if m`i'_sec_canon ==. &  m`i'_sec_sony == . &
m`i'_done == 1
* Select intervention time to be the minimum recorded time.  Note that sometimes
* the timers disagree by a second.
    gen m_sec`i' =  min(m`i'_sec_canon, m`i'_sec_sony)
}

* Check that we have an intervention time whenever the intervention was done
forvalues i = 1/10 {
    list id m_sec`i' if m_sec`i' == . & m`i'_done == 1
}

tempfile widerealdata
save "`widerealdata'", replace

* Determine a list of all of the unique times at which interventions occured in
* the first 15 minutes.  Set the first and last time to be 0 and 15 minutes,
* respectively.  These times will be used in all bootstrapped samples

* Reformat the data so that there is one record for each intervention
reshape long m_sec, i(id) j(intervention)
sort id m_sec
drop if m_sec == .

* convert time to minutes
gen m_min = m_sec / 60
label variable m_min "Time since start of scenario (min)"

* Drop data beyond 15 minutes
drop if m_min > 15
keep m_min
local obsplus2 = _N + 2
set obs `obsplus2'
replace m_min = 0 if _n == _N -1
replace m_min = 15 if _n == _N
sort m_min
drop if m_min == m_min[_n-1]

* We will use the rarea command to graph confidence intervals.  This command
* does not permit the stairstep connect option. To induce this connection we
* enter a time that is 0.01 minutes before each value of m_min if the gap between
* between consecutive values is greater than 0.01.

gen too_wide = 1 +(((m_min - m_min[_n-1] ) > 0.01 ) & (m_min[_n-1] != .))
expand too_wide
sort m_min
replace m_min = m_min[_n+1] -0.01 if m_min == m_min[_n+1]
global num_times = _N
di " times = $num_times"
tempfile times
save "`times'"

clear
use "`widerealdata'", clear
medianInterventionsPerformed `times' `n_teams'
di "program completed"
use "`times'", clear
gen n_times = _n
gen checkpercent = .
gen nocheckpercent = .
forvalues i = 1/$num_times {
    local timeWith r(timeWithCheck`i')
    local timeNo r(timeNoCheck`i')
    quietly replace checkpercent = ``timeWith'' if _n == `i'
    quietly replace nocheckpercent = ``timeNo'' if _n == `i'
}
list n_times m_min checkpercent nocheckpercent if m_min == 0 |      ///
    (m_min > 5.04 & m_min < 5.06) | (m_min > 10.1 & m_min < 10.21) | m_min == 15


twoway (line checkpercent m_min , connect(stairstep) lcolor(red) lwidth(medthick))   ///
       (line nocheckpercent m_min , connect(stairstep)lcolor(blue) lwidth(medthick)) ///
     , name(interventionByTime2) graphregion(fcolor(white) lcolor(white))            ///
       ylabel(0(10)100, angle(0))  xlabel(0(2)14) xmtick(0(1)15)                     ///
       ytitle(% of interventions performed)                                          ///
       legend(order(1 "With checklist" 2 "Without checklist")                        ///
           ring(0) position(11) col(1))
```

```
tempfile longrealdata
save "`longrealdata'", replace

use "`widerealdata'", clear
medianInterventionsPerformed `times' `n_teams'
clear
set obs 1
forvalues i = 1/$num_times {
    local timeWith r(timeWithCheck`i')
    local timeNo r(timeNoCheck`i')
    quietly gen timeWithCheck`i' = ``timeWith''
    quietly gen timeNoCheck`i' = ``timeNo''
}
gen n=0
preserve
drop timeNo*
reshape long timeWithCheck , i(n) j(time_pt)
gen checklist = 1
tempfile longrealwithforboot
save "`longrealwithforboot'"
restore
drop timeWith*
reshape long timeNoCheck , i(n) j(time_pt)
gen checklist = 0
tempfile longrealnoforboot
save "`longrealnoforboot'"

use "`widerealdata'", clear

* Bootstrap confidence intervals for the percent of interventions
* performed at each time, with or without the checklist.
* Note that the unit of selection in this bootstrap is the team id
* and not the team's performance at any time. That is, in each bootstrapped
* sample we either select (with replacement) either all or none of the
* observations for any specific id.

* Bootstrapping will be statified by checklist status.  That is, each bootstrapped sample will
* contain 38 teams that used checklists and 38 teams that do not.

local boot_arg = " "
forvalues i = 1/$num_times {
    local boot_arg = "`boot_arg' timeWithCheck`i' = r(timeWithCheck`i') timeNoCheck`i' =
r(timeNoCheck`i')"
}
di "`boot_arg'"
bootstrap `boot_arg', rep(2000)  saving(bootstrap.dta, replace) strata(checklist):   ///
    medianInterventionsPerformed `times' `n_teams'

use bootstrap.dta, replace
gen n =_n

preserve
drop timeNo*
reshape long timeWithCheck , i(n) j(time_pt)
gen checklist = 1
tempfile longwithboot
save   "`longwithboot'"
restore
drop timeWith*
reshape long timeNoCheck , i(n) j(time_pt)
gen checklist = 0
tempfile longnoboot
save   "`longnoboot'"

append using "`longwithboot'" "`longrealwithforboot'" "`longrealnoforboot'"
sort n checklist time_pt
tempfile longboot
save "`longboot'"
use "`times'", clear
sort m_min
gen time_pt = _n

save "`times'", replace
merge 1:m time_pt using "`longboot'"
sort n checklist m_min

* Replot the real data to see if I have messed up.
```

```
twoway (line timeWithCheck m_min if n ==0 & checklist==1, connect(stairstep) color(red) ///
    lwidth(medthick)) (line timeNoCheck m_min if n==0 & checklist == 0, color(blue)     ///
    lwidth(medthick) connect(stairstep))                                                ///
  , name(interventionbytimereal) graphregion(fcolor(white) lcolor(white))        ///
    ylabel(0(10)100, angle(0))  xlabel(0 (2) 14) xmtick(0(1)15) ytitle(% of interventions
performed)

* Plot each of the bootstrapped intervention percentages for checklist users as a function
* of time since start of intervention. Overlay the real intervention percentages on this
* plot.
preserve
keep if checklist == 1
twoway (line timeWithCheck m_min if n !=0 , connect(L) color(red) lwidth(vthin))      ///
     (line timeWithCheck m_min if n==0 & checklist == 1, color(blue) lwidth(medthick) ///
         connect(stairstep))                                                  ///
   , name(bootwithcheck) graphregion(fcolor(white) lcolor(white))                    ///
     ylabel(0(10)100, angle(0))  xlabel(0 (2) 14) xmtick(0(1)15)                     ///
     ytitle(% of interventions performed) legend(subtitle("With checklist")          ///
         order(1 "Bootstrapped samples" 2 "Real data") ring(0) position(10) cols(1))

restore

* Plot each of the bootstrapped intervention percentages for non-checklist users as a function
* of time since start of intervention. Overlay the real intervention percentages on this
* plot.

preserve
keep if checklist == 0
twoway (line timeNoCheck m_min if n !=0 , connect(L) color(red) lwidth(vthin))        ///
     (line timeNoCheck m_min if n==0 & checklist == 0, color(blue) lwidth(medthick)   ///
         connect(stairstep))                                                          ///
   , name(bootnocheck) graphregion(fcolor(white) lcolor(white))                    ///
     ylabel(0(10)100, angle(0))  xlabel(0 (2) 14) xmtick(0(1)15)                     ///
     ytitle(% of interventions performed) legend(subtitle("Without checklist")       ///
         order(1 "Bootstrapped samples" 2 "Real data") ring(0) position(10) cols(1))
restore

* Plot the true compliance  curve together with the bootstrapped
* 95% confidence interval bands based on bootstrapped percentiles.

keep if n==0
* e(ci_percentile) is a matrix returned by the bootstrap program that gives the percentile-
based
* 95% confidence intervals for each of the median number of interventions calculated by
* medianInterventionsPerformed
matrix CI_percentile = e(ci_percentile)
matrix list CI_percentile
di "$num_times"
sort checklist m_min
gen index = .
replace index =_n*2    if checklist == 0 & _n <= $num_times
replace index =(_n - $num_times)*2 -1 if checklist == 1 & _n > $num_times

gen lb_pct=CI_percentile[1,index]       if checklist == 0
gen ub_pct=CI_percentile[2,index]       if checklist == 0
replace lb_pct=CI_percentile[1,index]      if checklist == 1
replace ub_pct=CI_percentile[2,index]      if checklist == 1

twoway (rarea lb_pct ub_pct m_min if checklist ==1, color(red*.4) lwidth(none))       ///
       (rarea lb_pct ub_pct m_min if checklist ==0, color(blue*.4%50) lwidth(none))   ///
       (line timeWithCheck m_min ,connect(stairstep) color(red) lwidth(medthick))        ///
       (line timeNoCheck m_min ,connect(stairstep) color(blue) lwidth(medthick))      ///
     , name(InterventionsByTimeCIptile) graphregion(fcolor(white) lcolor(white))       ///
       ylabel(0(10)100, angle(0))  xlabel(0(2)14) xmtick(0(1)15 )                     ///
       ytitle(% of interventions performed)                                           ///
       legend(subtitle("95% CIs") order(1 "" 2 "")  ring(0) position(11) col(1))
log close
```

Appendix 1 Table 12

Appendix 1 Table 12 was created by the program *Table2.do*. It runs a number of mixed-effects ordinal logistic regression models (also known as proportional odds regression models) that assess the effects of checklists and other team attributes on the number of interventions performed in 15 minutes. Random intercepts for the EDs and treatment teams are included in these models, with the effect for teams nested within the effect for EDs. The program also performs a maximum likelihood ratio test to compare each mixed-effects model with the analogous fixed effects model. Your must run *Figure1.do* before running *Table2.do*. *Table2.do* is given in the following monospaced font.

```
log using Table2.log, replace
* Table2.log

* Explore mixed effects proportional odds models on 15 minute data

use Checklists_ED_interventions.dta, clear

* Calculate the observed % of interventions performed by teams that did,
* and did not, use checklists
tabulate n15_interventions checklist, col

* Regress interventions in 15 minutes against checklist usage using a
* proportion odds model with EDs and teams treated as random effects.
* Teams are nested within EDs in this model
meologit n15_interventions checklist || ed: || team:
display "z = " _b[checklist]/_se[checklist]

* Display the P value associated with the null hypothesis that checklists
* have no effect on the number of interventions performed.
display "P = "2*normal(-abs(_b[checklist]/_se[checklist]))

* Display probability that a team using the checklist performed each of the possible
* number of interventions.
lincom -_b[/cut1] + _b[checklist]
* Probability of 1 intervention with checklist
di 1- invlogit(r(estimate))
di 1- invlogit(r(lb))
di 1- invlogit(r(ub))

nlcom 1- invlogit(-_b[/cut1] + _b[checklist])
* Probability of 2 interventions with checklist
nlcom    invlogit(-_b[/cut1] + _b[checklist])- invlogit(-_b[/cut2] + _b[checklist])
* Probability of 3 interventions with checklist
nlcom    invlogit(-_b[/cut2] + _b[checklist])- invlogit(-_b[/cut3] + _b[checklist])
* Probability of 4 interventions with checklist
nlcom    invlogit(-_b[/cut3] + _b[checklist])- invlogit(-_b[/cut4] + _b[checklist])
* Probability of 5 interventions with checklist
nlcom    invlogit(-_b[/cut4] + _b[checklist])- invlogit(-_b[/cut5] + _b[checklist])
* Probability of 6 interventions with checklist
nlcom    invlogit(-_b[/cut5] + _b[checklist])- invlogit(-_b[/cut6] + _b[checklist])
* Probability of 7 interventions with checklist
nlcom    invlogit(-_b[/cut6] + _b[checklist])- invlogit(-_b[/cut7] + _b[checklist])
* Probability of 8 interventions with checklist
nlcom    invlogit(-_b[/cut7] + _b[checklist])- invlogit(-_b[/cut8] + _b[checklist])
* Probability of 9 interventions with checklist
nlcom    invlogit(-_b[/cut8] + _b[checklist])- invlogit(-_b[/cut9] + _b[checklist])
nlcom    invlogit(-_b[/cut9] + _b[checklist])
lincom    -_b[/cut9] + _b[checklist]
* Probability of 10 interventions with checklist
di invlogit(r(estimate))
di invlogit(r(lb))
di invlogit(r(ub))

* Display probability that a team not using the checklist performed each of the possible
* number of interventions.
lincom -_b[/cut1]
* Probability of 1 interventions without checklist
di 1- invlogit(r(estimate))
di 1- invlogit(r(lb))
di 1- invlogit(r(ub))

nlcom 1- invlogit(-_b[/cut1] )
* Probability of 2 interventions without checklist
nlcom    invlogit(-_b[/cut1] )- invlogit(-_b[/cut2] )
* Probability of 3 interventions without checklist
```

```
nlcom    invlogit(-_b[/cut2] )- invlogit(-_b[/cut3] )
* Probability of 4 interventions without checklist
nlcom    invlogit(-_b[/cut3] )- invlogit(-_b[/cut4] )
* Probability of 5 interventions without checklist
nlcom    invlogit(-_b[/cut4] )- invlogit(-_b[/cut5] )
* Probability of 6 interventions without checklist
nlcom    invlogit(-_b[/cut5] )- invlogit(-_b[/cut6] )
* Probability of 7 interventions without checklist
nlcom    invlogit(-_b[/cut6] )- invlogit(-_b[/cut7] )
* Probability of 8 interventions without checklist
nlcom    invlogit(-_b[/cut7] )- invlogit(-_b[/cut8] )
* Probability of 9 interventions without checklist
nlcom    invlogit(-_b[/cut8] )- invlogit(-_b[/cut9] )
nlcom    invlogit(-_b[/cut9] )
lincom    -_b[/cut9]
* Probability of 10 interventions without checklist
di invlogit(r(estimate))
di invlogit(r(lb))
di invlogit(r(ub))

* Investigate the importance of years of experience of the senior physician
tabulate p1_exp_years
meologit   n15_interventions i.p1_exp_years  || ed: || team:

* Investigate the importance of whether the senior physician is a specialist
tabulate p1_specialist
meologit   n15_interventions i.p1_specialist || ed: || team:

* Investigate the importance of whether the senior physician is a man
tabulate p1_male
meologit   n15_interventions i.p1_male || ed: || team:

* Investigate the importance of years of experience of the senior nurse
tabulate p2_exp_years
meologit   n15_interventions i.p2_exp_years || ed: || team:

* Investigate the importance of use of cognitive aids without checklist
gen cognitive_aid = uses_local_cognitive_aid
replace cognitive_aid = 2 if checklist == 1
tabulate cognitive_aid
meologit   n15_interventions i.cognitive_aid || ed: || team:
log close
```

Appendix 1 Table 13 and Table 14

Appendix 1 Table 13 and Table 14 were created by the program *Tables13&14.do*. It generates the P values for Table 13 and Table 14. Table 13 displays the impact of ED, team, senior physician experience, whether the senior physician is a specialist, scenario nurse, checklist access and scenario type on the number of interventions performed. Table 14 analyses whether any of these factors, with the addition of use of local cognitive aids, modifies the impact of checklist access on the number of interventions performed. These analyses also use mixed effects ordinal logistic regression models that are similar to those used in *Table2.do*.

```
log using Appendix1Tables13&14.log, replace
* Appendix1Tables13&14.log

* Test for confounding and interaction using a mixed effects proportional odds models on 15
minute data

use "..\Checklists_ED_interventions_short_200525.v5.dta", clear
tabulate n15_interventions checklist, col
gen col_n_int = n15_interventions
recode col_n_int 1/4 = 0 7/10 = 11
tabulate col_n_int checklist, col

* Test the importance of potentially confounding variables

* investigate the importance of individual EDs
meologit   n15_interventions i.ed   || team:
tabulate ed
test (2.ed 3.ed 4.ed)

* Investigate the importance of years of experience of the senior physician
tabulate p1_exp_years
meologit   n15_interventions i.p1_exp_years  || ed: || team:
test (2.p1_exp_years 3.p1_exp_years 4.p1_exp_years 5.p1_exp_years)

* Investigate the importance of whether the senior physician is a specialist
tabulate p1_specialist
meologit   n15_interventions i.p1_specialist || ed: || team:

* Investigate the importance of years of experience of the senior nurse
tabulate p2_exp_years
meologit   n15_interventions i.p2_exp_years || ed: || team:
test (2.p2_exp_years 3.p2_exp_years 4.p2_exp_years 5.p2_exp_years)

* Investigate the importance of checklist access
tabulate checklist
meologit   n15_interventions checklist || ed: || team:

* Investigate the importance of the scenario
tabulate scenario
meologit   n15_interventions i.scenario || ed: || team:
test (2.scenario 3.scenario 4.scenario 5.scenario 6.scenario 7.scenario 8.scenario )

* Test for interactions between checklists and potentially confounding variables

* investigate interaction between checklists and individual EDs
meologit   n15_interventions ed##checklist   || team:
test (2.ed#1.checklist 3.ed#1.checklist 4.ed#1.checklist  )

* Investigate interaction between checklists and years of experience of the senior physician
tabulate p1_exp_years
meologit   n15_interventions p1_exp_years##checklist  || ed: || team:
test (2.p1_exp_years#1.checklist 3.p1_exp_years#1.checklist 4.p1_exp_years#1.checklist
5.p1_exp_years#1.checklist)

* Investigate interaction between checklists and whether the senior physician is a specialist
tabulate p1_specialist
meologit   n15_interventions p1_specialist##checklist || ed: || team:

* Investigate interaction between checklists and years of experience of the senior nurse
tabulate p2_exp_years
meologit   n15_interventions p2_exp_years##checklist || ed: || team:
test (2.p2_exp_years#1.checklist 3.p2_exp_years#1.checklist 4.p2_exp_years#1.checklist
5.p2_exp_years#1.checklist)

* Investigate interaction between checklists and use of cognitive aids
tabulate uses_local_cognitive_aid
```

```
meologit   n15_interventions uses_local_cognitive_aid##checklist || ed: || team:

* Investigate interaction between checklists and the scenario
tabulate scenario
meologit   n15_interventions scenario##checklist || ed: || team:
test (2.scenario#1.checklist 3.scenario#1.checklist 4.scenario#1.checklist
5.scenario#1.checklist ///
    6.scenario#1.checklist 7.scenario#1.checklist 8.scenario#1.checklist )


log close
```