```
/*******************************************************************************
                          Semana 3: Problem Set 2

                          Universidad de San Andrés
                              Economía Aplicada
                                                               2022
>
*******************************************************************************/


* Source: https://www.aeaweb.org/articles?id=10.1257/app.20200204


/*******************************************************************************
Este archivo sigue la siguiente estructura:

0) Set up environment and globals

1) Regressions

*******************************************************************************/



* 0) Set up environment
*==============================================================================*

clear all
        version 16              // Set Version number for backward compatibility
        set more off            // Disable partitioned output
        clear all               // Start with a clean slate
        set linesize 80         // Line size limit to make output more readable

gl main "C:\Users\Anzony\Documents\GitHub\Applied_Econometrics\PS2"
gl input "$main/input"
gl output "$main/output"

* Open data set

use "$input/measures.dta", clear

* Global with control variables

global covs_eva "male i.eva_fu"
global covs_ent "male i.ent_fu"



* 1) Regressions
*==============================================================================*

******************************************************************************
* PANEL A (Child's cognitive skills at follow up)
******************************************************************************

        local bayley "b_tot_cog b_tot_lr b_tot_le b_tot_mf"
        local i = 1
        foreach y of local bayley{
        local append append
        if "`y'"=="b_tot_cog" local append replace
                cap drop V*
                reg `y'1_st treat `y'0_st $covs_eva , cluster(cod_dane)
                est sto panel_A_`i'
                local i = `i' + 1
                estadd scalar N1 = e(N)
        }
        local macarthur "mac_words mac_phrases"
        foreach y of local macarthur{
                cap drop V*
                reg `y'1_st treat mac_words0_st $covs_ent , cluster(cod_dane)
                est sto panel_A_`i'
                local i = `i' + 1
                estadd scalar N1 = e(N)
        }


*******************************************************************************
* PANEL B (Child's socio-emotional skills at follow up)
*******************************************************************************

        local bates "bates_difficult bates_unsociable bates_unstoppable"
        local i = 1
        foreach y of local bates{
                cap drop V*
                reg `y'1_st treat `y'0_st $covs_ent, cl(cod_dane)
                estadd scalar N1 = e(N)
                est sto panel_B_`i'
                local i = `i' + 1

        }

        local roth "roth_inhibit roth_attention"
        foreach y of local roth{
                cap drop V*
                reg `y'1_st treat bates_difficult0_st $covs_ent , cluster(cod_dane)
                estadd scalar N1 = e(N)
                est sto panel_B_`i'
                local i = `i' + 1
        }
        reg home_name1_st
        estadd scalar N1 = .
        est sto panel_B_6



*******************************************************************************
* PANEL C (Material investments)
*******************************************************************************

        local fcimat "fci_play_mat_type Npaintbooks Nthingsmove Ntoysshape Ntoysbought
> "
        local i = 1
        foreach y of local fcimat{
                cap drop V*
                reg `y'1_st treat fci_play_mat_type0_st $covs_ent , cluster(cod_dane)
                estadd scalar N1 = e(N)
                est sto panel_C_`i'
                local i = `i' + 1
        }
        reg home_name1_st
        estadd scalar N1 = .
        est sto panel_C_6



*******************************************************************************
* PANEL D (Time investments)
*******************************************************************************
        local fcitime "fci_play_act home_stories home_read home_toys home_name"
        local i = 1
        foreach y of local fcitime{
                cap drop V*
                reg `y'1_st treat fci_play_act0_st $covs_ent , cluster(cod_dane)
                estadd scalar N1 = e(N)
                est sto panel_D_`i'
                local i = `i' + 1

        }
        // aux reg for extra column
        reg home_name1_st
        estadd scalar N1 = .
        est sto panel_D_6


********************************************************************************
* Replicated Table
********************************************************************************


                * Making table for mortality
                #delimit ;

                        global note "Notes: All scores have been internally standardiz
> ed nonparametrically
                                        for age and are expressed in standard deviatio
> n
                                        units (see online Appendix B for details about
>
                                        the measures and the standardization procedure
> ).
                                        Measures followed by (-) have been reversed so
>  that a higher
                                        score refers to better behavior. The effects r
> elating
                                        to the latent factors are in log points. Coeff
> icients and
                                        standard errors clustered at the municipality
> level (in
                                        parentheses) are from a regression of the depe
> ndent variable
                                        measured at follow-up on an indicator for whet
> her
                                        the child received any psychosocial stimulatio
> n and controlling
                                        for the child’s sex, tester effects, and basel
> ine
                                        level of the outcome." ;

                #delimit cr


                # delimit ;

                        esttab panel_A_1 panel_A_2 panel_A_3
                        panel_A_4 panel_A_5 panel_A_6 using "${output}/table2_replicat
> ion.tex", replace
                        cells(b(label(coef.) star fmt(%8.3f) ) se(label((z)) par fmt(%
> 6.3f)))
                        starlevels(* 0.10 ** 0.05 *** 0.01)
                        s(N1 , label( "N" ) fmt(%9.0gc) )
                        collabels(none) nostar  noobs nonote
                         nonumbers eqlabels( none )
                        nonote
                        keep( treat )
                        varlabels( treat "Treatment"  )
                        mtitle(
                                "\shortstack{  \\ Bayley: \\ Cognitive}"
                                "\shortstack{  \\ Bayley: \\ Receptive language}"
                                "\shortstack{ \\ Bayley: \\ Expressive language}"
                                "\shortstack{  \\ Bayley: \\ Fine motor}"
                                "\shortstack{  \\ MacArthur: \\Words the child can say
> }"
                                "\shortstack{  \\ MacArthur: \\ Complex phrases \\ the
>  child can say}" )
                        mgroups( "\underline{ Panel A.} \textbf{ Child’s cognitive ski
> lls at follow-up }"
                                                                , pattern( 1 0 0 0 0 0
>  ) prefix(\multicolumn{@span}{c}{) suffix(}) span end(\hline) )
                        prehead("\begin{table} \small \centering
                                \protect \captionsetup{justification=centering}
                                \caption{\label{tab:table1} Treatment Impacts on Raw M
> easures and Latent Factors }"
                                "\noindent\resizebox{\textwidth}{!}{ \begin{threepartt
> able}"
                                "\begin{tabular}{lcccccc}" \toprule)
                        posthead(\hline) prefoot(\midrule) postfoot( \midrule) ;

                        esttab panel_B_1 panel_B_2 panel_B_3
                        panel_B_4 panel_B_5 panel_B_6  using "${output}/table2_replica
> tion.tex", append
                        cells(b(label(coef.) star fmt(%8.3f) ) se(label((z)) par fmt(%
> 6.3f)))
                        starlevels(* 0.10 ** 0.05 *** 0.01)
                        s(N1 , label( "N" ) fmt(%9.0gc) )
                        collabels( none )
                        nostar  noobs nonote
                         nonumbers eqlabels( none )
                        keep( treat )
                        mgroups( "
                                \underline{ Panel B.}
                                \textbf{ Child’s socio-emotional skills at follow-up }
> "
                                , pattern( 1 0 0 0 0 0 )
                                prefix(\multicolumn{@span}{c}{) suffix(})
                                span end(\hline) )
                        mtitle(
                                "\shortstack{ \\ ICQ: \\Difficult (-)}"
                                "\shortstack{ \\ ICQ:  \\ Unsociable (-)}"
                                "\shortstack{ \\ ICQ:  \\ Unstoppable (-)}"
                                "\shortstack{ \\ ECBQ: \\  Inhibitory control}"
                                "\shortstack{ \\ ECBQ: \\  Attentional focusing}" )
                        nonote
                        varlabels( treat "Treatment"  )
                        prehead( "" )
                         posthead( \hline )
                         prefoot(\midrule)
                         postfoot("")
                         delim("&") nonumbers
                          ;



                        esttab panel_C_1 panel_C_2 panel_C_3
                        panel_C_4 panel_C_5 panel_C_6  using "${output}/table2_replica
> tion.tex", append
                        cells(b(label(coef.) star fmt(%8.3f) ) se(label((z)) par fmt(%
> 6.3f)))
                        starlevels(* 0.10 ** 0.05 *** 0.01)
                        s(N1 , label( "N" ) fmt(%9.0gc) )
                        collabels(none) nostar  noobs
                        nonumbers eqlabels( none )
                        mgroups( "
                                \underline{ Panel C.}
                                \textbf{ Material investments at follow-up }"
                                , pattern( 1 0 0 0 0 0 )
                                prefix(\multicolumn{@span}{c}{) suffix(})
                                span end(\hline) )
                        mtitle(
                                "\shortstack{ \\ FCI: \\ Number of types\\ of play mat
> erials}"
                                "\shortstack{ \\ FCI: \\ Number of coloring \\ and dra
> wing books}"
                                "\shortstack{ \\ FCI: \\ Number of toys \\ to learn mo
> vement}"
                                "\shortstack{ \\ FCI: \\ Number of toys \\ to learn sh
> apes}"
                                "\shortstack{ \\ FCI: \\ Number of \\ shop-bought toys
> }"
                                "" )
                        nonote keep( treat )
                        varlabels( treat "Treatment"  )
```

```
                         prehead( \hline )
                         posthead( \hline )
                         prefoot(\midrule)
                         postfoot("")
                         delim("&") nonumbers
                          ;


                        esttab panel_D_1 panel_D_2 panel_D_3
                        panel_D_4 panel_D_5 panel_D_6  using "${output}/table2_replica
> tion.tex", append
                        cells(b(label(coef.) star fmt(%8.3f) ) se(label((z)) par fmt(%
> 6.3f)))
                        starlevels(* 0.10 ** 0.05 *** 0.01)
                        s(N1 , label( "N" ) fmt(%9.0gc) )
                        collabels(none) nostar  noobs nonote
                         nonumbers eqlabels( none ) keep( treat )
                        mgroups( "
                                \underline{ Panel D.}
                                \textbf{ Time investments at follow-up }"
                                , pattern( 1 0 0 0 0  )
                                prefix(\multicolumn{@span}{c}{) suffix(})
                                span end(\hline) )
                        mtitle(
                                "\shortstack{ \\ FCI: \\ Number of types\\ of play act
> ivities \\ in last 3 days}"
                                "\shortstack{ \\ FCI: \\ Number of times told \\ a sto
> ry to child \\ in last 3 days}"
                                "\shortstack{ \\ FCI: \\ Number of times read \\ to ch
> ild \\ in last 3 days}"
                                "\shortstack{ \\ FCI: \\ Number of times \\ played wit
> h toys \\ in last 3 days}"
                                "\shortstack{ \\ FCI: \\ Number of times \\ named thin
> gs to child \\ in last 3 days}"
                                "" )
                        varlabels( treat "Treatment"  )
                         prehead( \hline )
                         posthead( \hline )
                         prefoot(\midrule)
                         delim("&") nonumbers
                         postfoot( \hline \end{tabular}
                                \begin{tablenotes}
                                \begin{footnotesize}
                                ${note}
                                \end{footnotesize}
                                "\end{tablenotes} \end{threeparttable} } \end{table}")
>  ;

                #delimit cr


* 2) Modification
*==============================================================================*

* 2) P-Values Correction
*==============================================================================*


******************************************************************************
* PANEL A (Child's cognitive skills at follow up)
******************************************************************************

        * Define number of hypothesis
        scalar hyp = 6
        * Define level of significance
        scalar signif = 0.05
        local bayley "b_tot_cog b_tot_lr b_tot_le b_tot_mf"
        local i = 1
        local group_regressions = ""
        mat p_values = J(hyp,1,.)
        foreach y of local bayley{
        local append append
        if "`y'"=="b_tot_cog" local append replace
                cap drop V*
                reg `y'1_st treat `y'0_st $covs_eva , cluster(cod_dane)
                eststo panel_A_`i': test treat = 0
                mat p_values[`i',1]=r(p)
                scalar p_value = r(p)
                scalar corr_p_value = min(1,r(p)*hyp)
                estadd scalar bonferroni = corr_p_value
                estadd scalar N1 = e(N)

                local est_name panel_A_`i'
                local group_regressions `group_regressions' `est_name'

                local i = `i' + 1
        }

        local macarthur "mac_words mac_phrases"
        foreach y of local macarthur{
                cap drop V*
                reg `y'1_st treat mac_words0_st $covs_ent , cluster(cod_dane)
                eststo panel_A_`i': test treat = 0
                mat p_values[`i',1]=r(p)
                scalar p_value = r(p)
                scalar corr_p_value = min(1,r(p)*hyp)
                estadd scalar bonferroni = corr_p_value
                estadd scalar N1 = e(N)

                local est_name panel_A_`i'
                local group_regressions `group_regressions' `est_name'

                local i = `i' + 1
        }

                // Modification of p-vals
                preserve

                        // * Define number of hypothesis
                        // scalar hyp = hyp
                        // * Define level of significance
                        // scalar signif = signif
                        * Holm Correction
                                clear
                                // Bring the p_values matrix as column dta
                                svmat p_values
                                // Identify the original variable outcome
                                gen outcome_order_var = _n

                                // Sort values
                                sort p_values1
                                // Indicate the rank of the variable
                                gen rank = _n
                                // Identify if its significan or note
                                gen alpha_corr = signif/(hyp+1-rank)
                                gen significant_Holm = (p_values1<alpha_corr)
                                replace significant_Holm = 0 if significant_Holm[_n-1]
> ==0

                                // Sort again based on outcome order
                                sort outcome_order_var
                                // Export the result as a matrix
                                mkmat significant_Holm, matrix(holm_cor)
                                keep p_values1 outcome_order_var
                        * Benjamini et al.
                                rename (p_values1 outcome_order_var) (pval outcome)
                                quietly sum pval
                                local totalpvals = r(N)

                                * Sort the p-values in ascending order and generate a
> variable that codes each p-value's rank
                                quietly gen int original_sorting_order = _n
                                quietly sort pval
                                quietly gen int rank = _n if pval~=.

                                * Set the initial counter to 1
                                local qval = 1

                                * Generate the variable that will contain the BKY (200
> 6) sharpened q-values
                                gen bky06_qval = 1 if pval~=.

                                * Set up a loop that begins by checking which hypothes
> es are rejected at q = 1.000, then checks which hypotheses are rejected at q = 0.999
> , then checks which hypotheses are rejected at q = 0.998, etc.  The loop ends by che
> cking which hypotheses are rejected at q = 0.001.

                                local totalpvals = ${totalpvals}
                                local qval = 1
                                while `qval' > 0 {
                                        * First Stage
                                        * Generate the adjusted first stage q level we
>  are testing: q' = q/1+q
                                        local qval_adj = `qval'/(1+`qval')
                                        * Generate value q'*r/M
                                        gen fdr_temp1 = `qval_adj'*rank/`totalpvals'
                                        * Generate binary variable checking condition
> p(r) <= q'*r/M
                                        gen reject_temp1 = (fdr_temp1>=pval) if pval~=
> .
                                        * Generate variable containing p-value ranks f
> or all p-values that meet above condition
                                        gen reject_rank1 = reject_temp1*rank
                                        * Record the rank of the largest p-value that
> meets above condition
                                        egen total_rejected1 = max(reject_rank1)

                                        * Second Stage
                                        * Generate the second stage q level that accou
> nts for hypotheses rejected in first stage: q_2st = q'*(M/m0)
                                        local qval_2st = `qval_adj'*(`totalpvals'/(`to
> talpvals'-total_rejected1[1]))
                                        * Generate value q_2st*r/M
                                        gen fdr_temp2 = `qval_2st'*rank/`totalpvals'
                                        * Generate binary variable checking condition
> p(r) <= q_2st*r/M
                                        gen reject_temp2 = (fdr_temp2>=pval) if pval~=
> .
                                        * Generate variable containing p-value ranks f
> or all p-values that meet above condition
                                        gen reject_rank2 = reject_temp2*rank
                                        * Record the rank of the largest p-value that
> meets above condition
                                        egen total_rejected2 = max(reject_rank2)
                                        * A p-value has been rejected at level q if it
> s rank is less than or
                                        * equal to the rank of the max p-value that me
> ets the above condition
                                        replace bky06_qval = `qval' if rank <= total_r
> ejected2 & rank~=.
                                        * Reduce q by 0.001 and repeat loop
                                        drop fdr_temp* reject_temp* reject_rank* total
> _rejected*
                                        local qval = `qval' - .001
                                }


                                quietly sort original_sorting_order
                                pause off

                                mkmat bky06_qval, matrix(pval_bky06)


                        scalar i = 1
                        foreach reg_store of local group_regressions{

                                scalar holm_cor_val = holm_cor[i, 1]
                                scalar pval_bky06_val = pval_bky06[i, 1]

                                if holm_cor_val == 1 {
                                        est restore `reg_store'
                                        estadd local pholm "Significant"
                                        estadd scalar bky_06 = pval_bky06_val

                                }
                                if holm_cor_val == 0 {
                                        est restore `reg_store'
                                        estadd local pholm "No Significant"
                                        estadd scalar bky_06 = pval_bky06_val

                                }

                        }
                restore

*******************************************************************************
* PANEL B (Child's socio-emotional skills at follow up)
*******************************************************************************

        * Define number of hypothesis
        scalar hyp = 5
        * Define level of significance
        scalar signif = 0.05
        local bates "bates_difficult bates_unsociable bates_unstoppable"
        local i = 1
        local group_regressions = ""
        mat p_values = J(hyp,1,.)
        foreach y of local bates{
                cap drop V*
                reg `y'1_st treat `y'0_st $covs_ent, cl(cod_dane)
                eststo panel_B_`i': test treat = 0
                mat p_values[`i',1]=r(p)
                scalar p_value = r(p)
                scalar corr_p_value = min(1,r(p)*hyp)
                estadd scalar bonferroni = corr_p_value
                estadd scalar N1 = e(N)

                local est_name panel_B_`i'
                local group_regressions `group_regressions' `est_name'

                local i = `i' + 1

        }
        local roth "roth_inhibit roth_attention"
        foreach y of local roth{
                cap drop V*
                reg `y'1_st treat bates_difficult0_st $covs_ent , cluster(cod_dane)
                eststo panel_B_`i': test treat = 0
                mat p_values[`i',1]=r(p)
                scalar p_value = r(p)
                scalar corr_p_value = min(1,r(p)*hyp)
                estadd scalar bonferroni = corr_p_value
                estadd scalar N1 = e(N)

                local est_name panel_B_`i'
                local group_regressions `group_regressions' `est_name'

                local i = `i' + 1
        }
        reg home_name1_st
        estadd scalar N1 = .
        est sto panel_B_6


        // Modification of p-vals
                preserve

                        // * Define number of hypothesis
                        // scalar hyp = hyp
                        // * Define level of significance
                        // scalar signif = signif
                        * Holm Correction
                                clear
                                // Bring the p_values matrix as column dta
                                svmat p_values
                                // Identify the original variable outcome
                                gen outcome_order_var = _n

                                // Sort values
                                sort p_values1
                                // Indicate the rank of the variable
                                gen rank = _n
                                // Identify if its significan or note
                                gen alpha_corr = signif/(hyp+1-rank)
                                gen significant_Holm = (p_values1<alpha_corr)
                                replace significant_Holm = 0 if significant_Holm[_n-1]
> ==0

                                // Sort again based on outcome order
                                sort outcome_order_var
                                // Export the result as a matrix
                                mkmat significant_Holm, matrix(holm_cor)
                                keep p_values1 outcome_order_var


                        * Benjamini et al.
                                rename (p_values1 outcome_order_var) (pval outcome)
                                quietly sum pval
                                local totalpvals = r(N)

                                * Sort the p-values in ascending order and generate a
> variable that codes each p-value's rank
                                quietly gen int original_sorting_order = _n
                                quietly sort pval
                                quietly gen int rank = _n if pval~=.

                                * Set the initial counter to 1
                                local qval = 1

                                * Generate the variable that will contain the BKY (200
> 6) sharpened q-values
                                gen bky06_qval = 1 if pval~=.
                                * Set up a loop that begins by checking which hypothes
> es are rejected at q = 1.000, then checks which hypotheses are rejected at q = 0.999
> , then checks which hypotheses are rejected at q = 0.998, etc.  The loop ends by che
> cking which hypotheses are rejected at q = 0.001.

                                local totalpvals = ${totalpvals}
                                local qval = 1
                                while `qval' > 0 {
                                        * First Stage
                                        * Generate the adjusted first stage q level we
>  are testing: q' = q/1+q
                                        local qval_adj = `qval'/(1+`qval')
                                        * Generate value q'*r/M
                                        gen fdr_temp1 = `qval_adj'*rank/`totalpvals'
                                        * Generate binary variable checking condition
> p(r) <= q'*r/M
                                        gen reject_temp1 = (fdr_temp1>=pval) if pval~=
> .
                                        * Generate variable containing p-value ranks f
> or all p-values that meet above condition
                                        gen reject_rank1 = reject_temp1*rank
                                        * Record the rank of the largest p-value that
> meets above condition
                                        egen total_rejected1 = max(reject_rank1)

                                        * Second Stage
                                        * Generate the second stage q level that accou
> nts for hypotheses rejected in first stage: q_2st = q'*(M/m0)
                                        local qval_2st = `qval_adj'*(`totalpvals'/(`to
> talpvals'-total_rejected1[1]))
                                        * Generate value q_2st*r/M
                                        gen fdr_temp2 = `qval_2st'*rank/`totalpvals'
                                        * Generate binary variable checking condition
> p(r) <= q_2st*r/M
                                        gen reject_temp2 = (fdr_temp2>=pval) if pval~=
> .
                                        * Generate variable containing p-value ranks f
> or all p-values that meet above condition
                                        gen reject_rank2 = reject_temp2*rank
                                        * Record the rank of the largest p-value that
> meets above condition
                                        egen total_rejected2 = max(reject_rank2)

                                        * A p-value has been rejected at level q if it
> s rank is less than or
                                        * equal to the rank of the max p-value that me
> ets the above condition
                                        replace bky06_qval = `qval' if rank <= total_r
> ejected2 & rank~=.
                                        * Reduce q by 0.001 and repeat loop
                                        drop fdr_temp* reject_temp* reject_rank* total
> _rejected*
                                        local qval = `qval' - .001
                                }


                                quietly sort original_sorting_order
                                pause off

                                mkmat bky06_qval, matrix(pval_bky06)


                        scalar i = 1
                        foreach reg_store of local group_regressions{

                                scalar holm_cor_val = holm_cor[i, 1]
                                scalar pval_bky06_val = pval_bky06[i, 1]
                                if holm_cor_val == 1 {
                                        est restore `reg_store'
                                        estadd local pholm "Significant"
                                        estadd scalar bky_06 = pval_bky06_val

                                }
                                if holm_cor_val == 0 {
                                        est restore `reg_store'
                                        estadd local pholm "No Significant"
                                        estadd scalar bky_06 = pval_bky06_val

                                }

                        }
                restore

********************************************************************************
* PANEL C (Material investments)
********************************************************************************

        * Define number of hypothesis
        scalar hyp = 5
        * Define level of significance
        scalar signif = 0.05
        local fcimat "fci_play_mat_type Npaintbooks Nthingsmove Ntoysshape Ntoysbought
> "
        local i = 1
        local group_regressions = ""
        mat p_values = J(hyp,1,.)
        foreach y of local fcimat{
                cap drop V*
                reg `y'1_st treat fci_play_mat_type0_st $covs_ent , cluster(cod_dane)
                eststo panel_C_`i': test treat = 0
                mat p_values[`i',1]=r(p)
                scalar p_value = r(p)
                scalar corr_p_value = min(1,r(p)*hyp)
                estadd scalar bonferroni = corr_p_value
                estadd scalar N1 = e(N)

                local est_name panel_C_`i'
                local group_regressions `group_regressions' `est_name'

                local i = `i' + 1
        }
        reg home_name1_st
        estadd scalar N1 = .
        est sto panel_C_6

        // Modification of p-vals
                preserve

                        // * Define number of hypothesis
                        // scalar hyp = hyp
                        // * Define level of significance
                        // scalar signif = signif
                        * Holm Correction
                                clear
                                // Bring the p_values matrix as column dta
                                svmat p_values
                                // Identify the original variable outcome
                                gen outcome_order_var = _n

                                // Sort values
                                sort p_values1
                                // Indicate the rank of the variable
                                gen rank = _n
                                // Identify if its significan or note
                                gen alpha_corr = signif/(hyp+1-rank)
                                gen significant_Holm = (p_values1<alpha_corr)
                                replace significant_Holm = 0 if significant_Holm[_n-1]
> ==0
```

```
                                // Sort again based on outcome order
                                sort outcome_order_var
                                // Export the result as a matrix
                                mkmat significant_Holm, matrix(holm_cor)
                                keep p_values1 outcome_order_var


                        * Benjamini et al.
                                rename (p_values1 outcome_order_var) (pval outcome)
                                quietly sum pval
                                local totalpvals = r(N)

                                * Sort the p-values in ascending order and generate a
> variable that codes each p-value's rank
                                quietly gen int original_sorting_order = _n
                                quietly sort pval
                                quietly gen int rank = _n if pval~=.

                                * Set the initial counter to 1
                                local qval = 1

                                * Generate the variable that will contain the BKY (200
> 6) sharpened q-values
                                gen bky06_qval = 1 if pval~=.

                                * Set up a loop that begins by checking which hypothes
> es are rejected at q = 1.000, then checks which hypotheses are rejected at q = 0.999
> , then checks which hypotheses are rejected at q = 0.998, etc.  The loop ends by che
> cking which hypotheses are rejected at q = 0.001.

                                local totalpvals = ${totalpvals}
                                local qval = 1
                                while `qval' > 0 {
                                        * First Stage
                                        * Generate the adjusted first stage q level we
>  are testing: q' = q/1+q
                                        local qval_adj = `qval'/(1+`qval')
                                        * Generate value q'*r/M
                                        gen fdr_temp1 = `qval_adj'*rank/`totalpvals'
                                        * Generate binary variable checking condition
> p(r) <= q'*r/M
                                        gen reject_temp1 = (fdr_temp1>=pval) if pval~=
> .
                                        * Generate variable containing p-value ranks f
> or all p-values that meet above condition
                                        gen reject_rank1 = reject_temp1*rank
                                        * Record the rank of the largest p-value that
> meets above condition
                                        egen total_rejected1 = max(reject_rank1)

                                        * Second Stage
                                        * Generate the second stage q level that accou
> nts for hypotheses rejected in first stage: q_2st = q'*(M/m0)
                                        local qval_2st = `qval_adj'*(`totalpvals'/(`to
> talpvals'-total_rejected1[1]))
                                        * Generate value q_2st*r/M
                                        gen fdr_temp2 = `qval_2st'*rank/`totalpvals'
                                        * Generate binary variable checking condition
> p(r) <= q_2st*r/M
                                        gen reject_temp2 = (fdr_temp2>=pval) if pval~=
> .
                                        * Generate variable containing p-value ranks f
> or all p-values that meet above condition
                                        gen reject_rank2 = reject_temp2*rank
                                        * Record the rank of the largest p-value that
> meets above condition
                                        egen total_rejected2 = max(reject_rank2)
                                        * A p-value has been rejected at level q if it
> s rank is less than or
                                        * equal to the rank of the max p-value that me
> ets the above condition
                                        replace bky06_qval = `qval' if rank <= total_r
> ejected2 & rank~=.

                                        * Reduce q by 0.001 and repeat loop
                                        drop fdr_temp* reject_temp* reject_rank* total
> _rejected*

                                        local qval = `qval' - .001
                                }


                                quietly sort original_sorting_order
                                pause off

                                mkmat bky06_qval, matrix(pval_bky06)


                        scalar i = 1
                        foreach reg_store of local group_regressions{

                                scalar holm_cor_val = holm_cor[i, 1]
                                scalar pval_bky06_val = pval_bky06[i, 1]

                                if holm_cor_val == 1 {
                                        est restore `reg_store'
                                        estadd local pholm "Significant"
                                        estadd scalar bky_06 = pval_bky06_val

                                }
                                if holm_cor_val == 0 {
                                        est restore `reg_store'
                                        estadd local pholm "No Significant"
                                        estadd scalar bky_06 = pval_bky06_val

                                }

                        }
                restore

********************************************************************************
* PANEL D (Time investments)
********************************************************************************

        * Define number of hypothesis
        scalar hyp = 5
        * Define level of significance
        scalar signif = 0.05
        local fcitime "fci_play_act home_stories home_read home_toys home_name"
        local i = 1
        local group_regressions = ""
        mat p_values = J(hyp,1,.)
        foreach y of local fcitime{
                cap drop V*
                reg `y'1_st treat fci_play_act0_st $covs_ent , cluster(cod_dane)
                eststo panel_D_`i': test treat = 0
                mat p_values[`i',1]=r(p)
                scalar p_value = r(p)
                scalar corr_p_value = min(1,r(p)*hyp)
                estadd scalar bonferroni = corr_p_value
                estadd scalar N1 = e(N)

                local est_name panel_D_`i'
                local group_regressions `group_regressions' `est_name'
                local i = `i' + 1

        }
        // aux reg for extra column
        reg home_name1_st
        estadd scalar N1 = .
        est sto panel_D_6
        // Modification of p-vals
                preserve

                        // * Define number of hypothesis
                        // scalar hyp = hyp
                        // * Define level of significance
                        // scalar signif = signif
                        * Holm Correction
                                clear
                                // Bring the p_values matrix as column dta
                                svmat p_values
                                // Identify the original variable outcome
                                gen outcome_order_var = _n

                                // Sort values
                                sort p_values1
                                // Indicate the rank of the variable
                                gen rank = _n
                                // Identify if its significan or note
                                gen alpha_corr = signif/(hyp+1-rank)
                                gen significant_Holm = (p_values1<alpha_corr)
                                replace significant_Holm = 0 if significant_Holm[_n-1]
> ==0


                                // Sort again based on outcome order
                                sort outcome_order_var
                                // Export the result as a matrix
                                mkmat significant_Holm, matrix(holm_cor)
                                keep p_values1 outcome_order_var


                        * Benjamini et al.
                                rename (p_values1 outcome_order_var) (pval outcome)
                                quietly sum pval
                                local totalpvals = r(N)

                                * Sort the p-values in ascending order and generate a
> variable that codes each p-value's rank
                                quietly gen int original_sorting_order = _n
                                quietly sort pval
                                quietly gen int rank = _n if pval~=.

                                * Set the initial counter to 1
                                local qval = 1

                                * Generate the variable that will contain the BKY (200
> 6) sharpened q-values
                                gen bky06_qval = 1 if pval~=.

                                * Set up a loop that begins by checking which hypothes
> es are rejected at q = 1.000, then checks which hypotheses are rejected at q = 0.999
> , then checks which hypotheses are rejected at q = 0.998, etc.  The loop ends by che
> cking which hypotheses are rejected at q = 0.001.
```

```
                                local totalpvals = ${totalpvals}
                                local qval = 1
                                while `qval' > 0 {
                                        * First Stage
                                        * Generate the adjusted first stage q level we
>  are testing: q' = q/1+q
                                        local qval_adj = `qval'/(1+`qval')
                                        * Generate value q'*r/M
                                        gen fdr_temp1 = `qval_adj'*rank/`totalpvals'
                                        * Generate binary variable checking condition
> p(r) <= q'*r/M
                                        gen reject_temp1 = (fdr_temp1>=pval) if pval~=
> .
                                        * Generate variable containing p-value ranks f
> or all p-values that meet above condition
                                        gen reject_rank1 = reject_temp1*rank
                                        * Record the rank of the largest p-value that
> meets above condition
                                        egen total_rejected1 = max(reject_rank1)

                                        * Second Stage
                                        * Generate the second stage q level that accou
> nts for hypotheses rejected in first stage: q_2st = q'*(M/m0)
                                        local qval_2st = `qval_adj'*(`totalpvals'/(`to
> talpvals'-total_rejected1[1]))
                                        * Generate value q_2st*r/M
                                        gen fdr_temp2 = `qval_2st'*rank/`totalpvals'
                                        * Generate binary variable checking condition
> p(r) <= q_2st*r/M
                                        gen reject_temp2 = (fdr_temp2>=pval) if pval~=
> .
                                        * Generate variable containing p-value ranks f
> or all p-values that meet above condition
                                        gen reject_rank2 = reject_temp2*rank
                                        * Record the rank of the largest p-value that
> meets above condition
                                        egen total_rejected2 = max(reject_rank2)

                                        * A p-value has been rejected at level q if it
> s rank is less than or
                                        * equal to the rank of the max p-value that me
> ets the above condition
                                        replace bky06_qval = `qval' if rank <= total_r
> ejected2 & rank~=.
                                        * Reduce q by 0.001 and repeat loop
                                        drop fdr_temp* reject_temp* reject_rank* total
> _rejected*
                                        local qval = `qval' - .001
                                }


                                quietly sort original_sorting_order
                                pause off

                                mkmat bky06_qval, matrix(pval_bky06)


                        scalar i = 1
                        foreach reg_store of local group_regressions{

                                scalar holm_cor_val = holm_cor[i, 1]
                                scalar pval_bky06_val = pval_bky06[i, 1]
                                if holm_cor_val == 1 {
                                        est restore `reg_store'
                                        estadd local pholm "Significant"
                                        estadd scalar bky_06 = pval_bky06_val

                                }
                                if holm_cor_val == 0 {
                                        est restore `reg_store'
                                        estadd local pholm "No Significant"
                                        estadd scalar bky_06 = pval_bky06_val

                                }

                        }
                restore

********************************************************************************
* Replicated Table
********************************************************************************


                * Making table for mortality
                #delimit ;

                        global note "Notes: All scores have been internally standardiz
> ed nonparametrically
                                        for age and are expressed in standard deviatio
> n
                                        units (see online Appendix B for details about
>
                                        the measures and the standardization procedure
> ).
                                        Measures followed by (-) have been reversed so
>  that a higher
                                        score refers to better behavior. The effects r
> elating
                                        to the latent factors are in log points. Coeff
> icients and
                                        standard errors clustered at the municipality
> level (in
                                        parentheses) are from a regression of the depe
> ndent variable
                                        measured at follow-up on an indicator for whet
> her
                                        the child received any psychosocial stimulatio
> n and controlling
                                        for the child's sex, tester effects, and basel
> ine
                                        level of the outcome." ;

                #delimit cr


                # delimit ;

                        esttab panel_A_1 panel_A_2 panel_A_3
                        panel_A_4 panel_A_5 panel_A_6 using "${output}/table2_replicat
> ion_newpval.tex", replace
                        cells(b(label(coef.) star fmt(%8.3f) ) se(label((z)) par fmt(%
> 6.3f)))
                        starlevels(* 0.10 ** 0.05 *** 0.01)
                        s(N1 bonferroni pholm bky_06,  label( "N" "Bonferroni P-val" "
> Holm P-value" "BKY P-val") fmt(%9.0gc) )
                        collabels(none) nostar  noobs nonote
                         nonumbers eqlabels( none )
                        nonote
                        keep( treat )
                        varlabels( treat "Treatment"  )
                        mtitle(
                                "\shortstack{  \\ Bayley: \\ Cognitive}"
                                "\shortstack{  \\ Bayley: \\ Receptive language}"
                                "\shortstack{ \\ Bayley: \\ Expressive language}"
                                "\shortstack{  \\ Bayley: \\ Fine motor}"
                                "\shortstack{  \\ MacArthur: \\Words the child can say
> }"
                                "\shortstack{  \\ MacArthur: \\ Complex phrases \\ the
>  child can say}" )
                        mgroups( "\underline{ Panel A.} \textbf{ Child’s cognitive ski
> lls at follow-up }"
                                                                , pattern( 1 0 0 0 0 0
>  ) prefix(\multicolumn{@span}{c}{) suffix(}) span end(\hline) )
                        prehead("\begin{table} \small \centering
                                \protect \captionsetup{justification=centering}
                                \caption{\label{tab:table1} Treatment Impacts on Raw M
> easures and Latent Factors - New Inference Value }"
                                "\noindent\resizebox{\textwidth}{!}{ \begin{threepartt
> able}"
                                "\begin{tabular}{lcccccc}" \toprule)
                        posthead(\hline) prefoot(\midrule) postfoot( \midrule) ;


                        esttab panel_B_1 panel_B_2 panel_B_3
                        panel_B_4 panel_B_5 panel_B_6  using "${output}/table2_replica
> tion_newpval.tex", append
                        cells(b(label(coef.) star fmt(%8.3f) ) se(label((z)) par fmt(%
> 6.3f)))
                        starlevels(* 0.10 ** 0.05 *** 0.01)
                        s( N1 bonferroni pholm bky_06,
                                label( "N" "Bonferroni P-val" "Holm P-value" "BKY P-va
> l")
                                fmt(0 3 3 ) )
                        collabels( none )
                        nostar  noobs nonote
                         nonumbers eqlabels( none )
                        keep( treat )
                        mgroups( "
                                \underline{ Panel B.}
                                \textbf{ Child’s socio-emotional skills at follow-up }
> "
                                , pattern( 1 0 0 0 0 0 )
                                prefix(\multicolumn{@span}{c}{) suffix(})
                                span end(\hline) )
                        mtitle(
                                "\shortstack{ \\ ICQ: \\Difficult (-)}"
                                "\shortstack{ \\ ICQ:  \\ Unsociable (-)}"
                                "\shortstack{ \\ ICQ:  \\ Unstoppable (-)}"
                                "\shortstack{ \\ ECBQ: \\  Inhibitory control}"
                                "\shortstack{ \\ ECBQ: \\  Attentional focusing}" )
                        nonote
                        varlabels( treat "Treatment"  )
                        prehead( "" )
                         posthead( \hline )
                         prefoot(\midrule)
                         postfoot("")
                         delim("&") nonumbers
                          ;




                        esttab panel_C_1 panel_C_2 panel_C_3
                        panel_C_4 panel_C_5 panel_C_6  using "${output}/table2_replica
> tion_newpval.tex", append
                        cells(b(label(coef.) star fmt(%8.3f) ) se(label((z)) par fmt(%
> 6.3f)))
                        starlevels(* 0.10 ** 0.05 *** 0.01)
                        s( N1 bonferroni pholm bky_06,
                                label( "N" "Bonferroni P-val" "Holm P-value" "BKY P-va
> l")
                                fmt(0 3 3 ) )
                        collabels(none) nostar  noobs
                        nonumbers eqlabels( none )
                        mgroups( "
                                \underline{ Panel C.}
                                \textbf{ Material investments at follow-up }"
                                , pattern( 1 0 0 0 0 0 )
                                prefix(\multicolumn{@span}{c}{) suffix(})
                                span end(\hline) )
                        mtitle(
                                "\shortstack{ \\ FCI: \\ Number of types\\ of play mat
> erials}"
                                "\shortstack{ \\ FCI: \\ Number of coloring \\ and dra
> wing books}"
                                "\shortstack{ \\ FCI: \\ Number of toys \\ to learn mo
> vement}"
                                "\shortstack{ \\ FCI: \\ Number of toys \\ to learn sh
> apes}"
                                "\shortstack{ \\ FCI: \\ Number of \\ shop-bought toys
> }"
                                "" )
                        nonote keep( treat )
                        varlabels( treat "Treatment"  )
                         prehead( \hline )
                         posthead( \hline )
                         prefoot(\midrule)
                         postfoot("")
                         delim("&") nonumbers
                          ;


                        esttab panel_D_1 panel_D_2 panel_D_3
                        panel_D_4 panel_D_5 panel_D_6  using "${output}/table2_replica
> tion_newpval.tex", append
                        cells(b(label(coef.) star fmt(%8.3f) ) se(label((z)) par fmt(%
> 6.3f)))
                        starlevels(* 0.10 ** 0.05 *** 0.01)
                        s( N1 bonferroni pholm bky_06,
                                label( "N" "Bonferroni P-val" "Holm P-value" "BKY P-va
> l")
                                fmt(0 3 3 ) )
                        collabels(none) nostar  noobs nonote
                         nonumbers eqlabels( none ) keep( treat )
                        mgroups( "
                                \underline{ Panel D.}
                                \textbf{ Time investments at follow-up }"
                                , pattern( 1 0 0 0 0  )
                                prefix(\multicolumn{@span}{c}{) suffix(})
                                span end(\hline) )
                        mtitle(
                                "\shortstack{ \\ FCI: \\ Number of types\\ of play act
> ivities \\ in last 3 days}"
                                "\shortstack{ \\ FCI: \\ Number of times told \\ a sto
> ry to child \\ in last 3 days}"
                                "\shortstack{ \\ FCI: \\ Number of times read \\ to ch
> ild \\ in last 3 days}"
                                "\shortstack{ \\ FCI: \\ Number of times \\ played wit
> h toys \\ in last 3 days}"
                                "\shortstack{ \\ FCI: \\ Number of times \\ named thin
> gs to child \\ in last 3 days}"
                                "" )
                        varlabels( treat "Treatment"  )
                         prehead( \hline )
                         posthead( \hline )
                         prefoot(\midrule)
                         delim("&") nonumbers
                         postfoot( \hline \end{tabular}
                                \begin{tablenotes}
                                \begin{footnotesize}
                                ${note}
                                \end{footnotesize}
                                "\end{tablenotes} \end{threeparttable} } \end{table}")
>  ;

                #delimit cr
```