Supplemental Table S1. Changes in Model Variables

			Original Variable						New Variable			
			Baseline	Year 1	Year 2	Year 3			Baseline	Year 1	Year 2	Year 3
Variable	Response Category		N (%)				Response Category		N (%)			
Establishing rules	1	Not allowed to use marijuana under any circumstances	7,272 (74.20%)	7121 (77.30%)	7,328 (81.89%)	7,072 (85.70%)	5-6	No rules	2,486 (24.48%)	2,054 (22.39%)	1,551 (17.47%)	1,122 (13.69%)
	2	Not allowed to use marijuana in home - no family rules outside home	33 (0.34%)	33 (0.36%)	50 (0.56%)	47 (0.57%)	1	Zero tolerance rules	7,272 (74.52%)	7,121 (77.61%)	7,325 (82.53%)	7,071 (86.31%)
	3	Allowed to use marijuana in home with permission	8 (0.08%)	3 (0.03%)	18 (0.20%)	10 (0.12%)		-				
	4	Allowed to use marijuana in home whenever he/she wants	1 (0.01%)	1 (0.01%)	1 (0.01%)	1 (0.01%)		-				
	5	I don't set rules about marijuana use	52 (0.53%)	62 (0.67%)	58 (0.65%)	68 (0.82%)		-				
	6	I haven't made rules yet about my child's marijuana use	2,434 (24.84%)	1992 (21.62%)	1,494 (16.69%)	1054 (12.77%)		-				
Enforcing rules	1	No, applies only to children, not teen or adults	136 (1.86%)	119 (1.66%)	145 (1.96%)	123 (1.73%)	1-2	Enforced for youth only	1,272 (17.39%)	1,407 (19.66%)	1,751 (23.68%)	1,788 (25.08%)
	2	No, applies only to children and teen, not adults	1,136 (15.53%)	1,288 (17.99%)	1,606 (21.71%)	1,664 (23.34%)	3	Enforced for everyone	6,042 (82.61%)	5,751 (80.34%)	5,646 (76.33%)	5,343 (74.94%)
	3	Yes	6,042 (82.61%)	5,751 (80.34%)	5,646 (76.33%)	5,343 (74.94%)		-				
Cannabis access	1	Very hard	8,340 (85.10%)	7,654 (83.09%)	6,974 (77.93%)	6,376 (77.27%)	1	Very hard	8,340 (85.10%)	7,654 (83.09%)	6,974 (77.93%)	6,376 (77.27%)
	2	Sort of hard	524 (5.35%)	689 (7.48%)	913 (10.20%)	960 (11.63%)	2-4	Sort of hard to very easy	815 (8.32%)	1,008 (10.94%)	1,392 (15.56%)	1,417 (17.19%)
	3	Sort of easy	195 (1.99%)	217 (2.36%)	335 (3.74%)	339 (4.11%)	5	Don't know	645 (6.58%)	550 (5.97%)	581 (6.50%)	458 (5.55%)

4	Very easy	96 (0.98%)	102 (1.11%)	145 (1.62%)	119 (1.44%)	-
5	Don't know	645 (6.58%)	550 (5.97%)	581 (6.50%)	458 (5.55%)	-

Note. Establishing rules – No rules = original response categories 5 and 6, zero tolerance rules = original category 1; Enforcing rules – Enforced for youth only = original response categories 1 and 2, Enforced for everyone = original response category 3, original response categories 2-4 were dropped and not included in the new variable; Cannabis access – Very hard = original response category 1, sort of hard to very easy = original response categories 2-4, Don't know = original response category 3.

Supplemental Table S2. Multiple Imputation Descriptive Statistics

	Non-Imputed				Imputed			
Variable	*Mean*	*SD*	*Skew*	*Kurtosis*	*Mean*	*SD*	*Skew*	*Kurtosis*
Establish Cannabis Rules								
Baseline	0.75	0.44	-1.13	-0.73	0.75	0.44	-1.13	-0.73
Y1	0.78	0.42	-1.33	-0.24	0.78	0.42	-1.34	-0.21
Y2	0.83	0.38	-1.71	0.94	0.83	0.38	-1.73	1.00
Y3	0.86	0.34	-2.11	2.46	0.87	0.34	-2.14	2.58
Enforce Cannabis Rules								
Baseline	0.83	0.38	-1.72	0.96	0.83	0.38	-1.73	0.98
Y1	0.80	0.40	-1.53	0.33	0.80	0.40	-1.52	0.32
Y2	0.76	0.43	-1.24	-0.47	0.76	0.43	-1.23	-0.49
Y3	0.75	0.43	-1.15	-0.68	0.74	0.44	-1.11	-0.76
Legal Status	2.12	0.69	-0.16	-0.88	2.12	0.69	-0.16	-0.88
Perceived Cannabis Access								
Baseline	0.21	0.55	2.47	4.79	0.21	0.55	2.47	4.78
Y1	0.23	0.54	2.31	4.18	0.23	0.54	2.32	4.24
Y2	0.29	0.58	1.90	2.45	0.28	0.58	1.90	2.46
Y3	0.28	0.56	1.86	2.41	0.28	0.56	1.87	2.45
Parental Monitoring								
Baseline	4.39	0.52	-1.16	1.70	4.39	0.52	-1.16	1.69
Y1	4.49	0.45	-1.28	2.44	4.48	0.46	-1.27	2.37
Y2	4.48	0.47	-1.34	2.70	4.48	0.47	-1.34	2.68
Y3	4.38	0.50	-0.95	1.03	4.37	0.50	-0.95	1.05
Parent Education	2.60	1.27	-0.15	-0.47	2.60	1.28	-0.15	-0.47
Household Income	7.18	2.45	-1.02	0.14	7.06	2.51	-0.94	-0.07
Child Sex	1.48	0.50	0.10	-1.99	1.48	0.50	0.10	-1.99

Note. Multiple imputation by chained equations (MICE) was implemented using the mice package in R (version 4.5.1). A total of 20 datasets were imputed.

Supplemental Table S3. GEE Logistic Models for No Change in Caregivers

		Establish Cannabis Rules		Enforce Cannabis Rules	
Variable	*Comparison*	*OR*	*95% CI*	*OR*	*95% CI*
Legal Status	General Adult Use vs. Medical	1.04	(0.95 – 1.15)	0.62	(0.56 – 0.69)
	General Adult Use vs. Low THC/CBD	0.68	(0.61 – 0.76)	0.39	(0.34 – 0.44)
	Medical vs. Low THC/CBD	0.65	(0.60 – 0.72)	0.62	(0.56 – 0.69)
Perceived Access	Easy vs. Hard	0.67	(0.62 – 0.73)	0.46	(0.42 – 0.50)
	Don't Know vs. Hard	0.58	(0.52 – 0.64)	0.83	(0.74 – 0.94)
	Don't Know vs. Easy	0.86	(0.77 – 0.97)	0.55	(0.49 – 0.62)
Parental Monitoring	-	0.98	(0.92 – 1.05)	1.04	(0.98 – 1.11)
Parent Race & Ethnicity	White, non-His	*ref*			
	NA/NH/PI	1.31	(0.93 – 1.85)	0.93	(0.65 – 1.34)
	Black, mono	2.13	(1.83 – 2.48)	0.81	(0.71 – 0.93)
	Black, multi	1.71	(1.06 – 2.76)	0.68	(0.46 – 1.01)
	His, multi	1.53	(1.25 – 1.86)	1.73	(1.38 – 2.17)
	His, White	1.27	(1.10 – 1.46)	1.93	(1.62 – 2.29)
	Asian, multi	1.09	(0.89 – 1.32)	1.44	(1.12 – 1.85)
Parent Education	Some HS	*ref*			
	HS or GED	0.94	(0.75 – 1.17)	0.84	(0.68 – 1.03)
	Associate's	0.78	(0.63 – 0.96)	0.74	(0.61 – 0.91)
	Bachelor's	0.59	(0.47 – 0.73)	0.98	(0.78 – 1.24)
	Master's	0.49	(0.39 – 0.61)	0.88	(0.69 – 1.12)
	Doctoral/Professional	0.59	(0.45 – 0.76)	1.01	(0.75 – 1.36)
Household Income	<$5,000	*ref*			
	$5,000-$11,999	0.92	(0.65 – 1.29)	1.17	(0.89 – 1.55)
	$12,000-$15,999	1.07	(0.73 – 1.56)	1.15	(0.84 – 1.57)
	$16,000-$24,999	0.76	(0.55 – 1.04)	1.10	(0.80 – 1.52)
	$25,000-$34,999	0.67	(0.50 – 0.90)	1.08	(0.82 – 1.42)
	$35,000-$49,999	0.76	(0.57 – 1.02)	1.22	(0.92 – 1.61)
	$50,000-$74,999	0.85	(0.65 – 1.13)	1.21	(0.92 – 1.61)
	$75,000-$99,000	0.84	(0.64 – 1.10)	1.39	(1.05 – 1.84)
	$100,000-$199,999	0.86	(0.65 – 1.12)	1.29	(0.98 – 1.71)
	>$200,000	0.93	(0.70 – 1.24)	1.26	(0.93 – 1.71)
Child Sex	Male vs. Female	0.99	(0.91 – 1.07)	1.05	(0.95 – 1.15)

Timepoint	-	1.32	(1.29 – 1.35)	0.88	(0.86 – 0.89)

Note. Statistically significant effects do not include 0 in the confidence limit. OR depicts the likelihood of establishing rules and enforcing rules for everyone, respectively. Multi = multiracial (more than one race), Mono = monoracial (only one race), His = Hispanic, NA/NH/PI = Native American/Native Hawaiian/Pacific Islander. OR = Odds ratio. CI = confidence interval.

Supplemental Materials 1 – SAS Syntax

SAS software version 9.4 (SAS Institute Inc., 2023)

Establish Rules Model

Step 1: Estimate models per imputation

```
ods trace on;

proc genmod data=can_final_MI1 descending;

class src_subject_id can_any_rules can_legal0 c_sex0 p_race0 can_access par_edu0
com_inc0;

model can_rule_establish = can_legal0 can_access par_mon p_race0 par_edu0 com_inc0
c_sex0 timepoint / DIST=BIN LINK=LOGIT covb;

repeated sub=src_subject_id / type=AR(1);

lsmeans can_legal0  / pdiff exp cl ilink;

*lsmeans can_access / pdiff exp cl ilink;

*lsmeans p_race0 / pdiff exp cl ilink;

by _imputation_;

ods output GEEEmpPEst=param_estimates CovB=param_cov lsmeans=lsm_out_establish
diffs= can_legal0; run;

ods trace off;

*each LSMEANS comparison must be done one at a time ;
```

Step 2: Pool estimates across imputations

```
*pooling estimates across imputations - establishing rules;

PROC MIANALYZE parms(classvar=level)=param_estimates;

class can_legal0 c_sex0 p_race0 can_access par_edu0 com_inc0;
```

```
MODELEFFECTS INTERCEPT can_legal0 can_access par_mon p_race0 par_edu0
com_inc0 c_sex0 timepoint;

run;
```

Step 3: After running LSMEANS statement per predictor (can_legal0, can_access, p_race0), pool comparisons of different levels of predictors.

```
*cannabis legal status----------;

*creating dummy comparison variable to allow for comparing different levels of legal status;

data can_legal0_d;

   set can_legal0_d;

   comparison = cats("Group", can_legal0, "_vs_", _can_legal0);

run;

*double checking name of comparison group was correct;

proc print data=can_legal0_d (obs=10); run;

*ensuring each comparison uses multiple imputations and sorting by ascending order;

proc sort data=can_legal0_d; by comparison _imputation_; run;


proc freq data=can_legal0_d noprint;

   tables comparison / out=comp_counts;

run;


data keep_comparisons;

   set comp_counts;

   if count >= 2;

run;


proc sql;

   create table can_legal0_d_filtered as

   select a.*

   from can_legal0_d a

   inner join keep_comparisons b
```

```
    on a.comparison = b.comparison;
quit;

*sorting by ascending order of comparison variable;
proc sort data=can_legal0_d_filtered; by comparison; run;

*pooling comparisons of different levels of legal status;
proc mianalyze data=can_legal0_d_filtered;
    by comparison;
    modeleffects Estimate;
    stderr StdErr;
run;

*cannabis access----------;
*creating dummy comparison variable to allow for comaparing different levels of cannabis access;
data can_access_d;
    set can_access_d;
    comparison = cats("Group", can_access, "_vs_", _can_access);
run;
*double checking name of comparison group was correct;
proc print data=can_access_d (obs=10); run;
*ensuring each comparison uses multiple imputations and sorting by ascending order;
proc sort data=can_access_d; by comparison _imputation_; run;

proc freq data=can_access_d noprint;
    tables comparison / out=comp_counts;
run;
```

```
data keep_comparisons;
    set comp_counts;
    if count >= 2;
run;

proc sql;
    create table can_access_d_filtered as
    select a.*
    from can_access_d a
    inner join keep_comparisons b
    on a.comparison = b.comparison;
quit;

*sorting by ascending order of comparison variable;
proc sort data=can_access_d_filtered; by comparison; run;

*pooling comparisons of different levels of cannabis access;
proc mianalyze data=can_access_d_filtered;
    by comparison;
    modeleffects Estimate;
    stderr StdErr;
run;

*parent race----------;
*creating dummy comparison variable to allow for comaparing different levels of race;
data p_race0_d;
    set p_race0_d;
    comparison = cats("Group", p_race0, "_vs_", _p_race0);
run;
```

```
*double checking name of comparison group was correct;
proc print data=p_race0_d (obs=10); run;
*ensuring each comparison uses multiple imputations and sorting by ascending order;
proc sort data=p_race0_d; by comparison _imputation_; run;

proc freq data=p_race0_d noprint;
   tables comparison / out=comp_counts;
run;

data keep_comparisons;
   set comp_counts;
   if count >= 2;
run;

proc sql;
   create table p_race0_d_filtered as
   select a.*
   from p_race0_d a
   inner join keep_comparisons b
   on a.comparison = b.comparison;
quit;

*sorting by ascending order of comparison variable;
proc sort data=p_race0_d_filtered; by comparison; run;

*pooling comparisons of different levels of race;
proc mianalyze data=p_race0_d_filtered;
   by comparison;
   modeleffects Estimate;
```

```
   stderr StdErr;
run;
```

Enforce Rules Model

Step 1: Estimate models per imputation

```
ods trace on;
proc genmod data=can_final_MI1 descending;
class src_subject_id can_any_rules can_legal0 c_sex0 p_race0 can_access par_edu0
com_inc0;
model can_rule_enforce = can_legal0 can_access par_mon p_race0 par_edu0 com_inc0
c_sex0 timepoint / DIST=BIN LINK=LOGIT covb;
repeated sub=src_subject_id / type=AR(1);
lsmeans can_legal0  / pdiff exp cl ilink;
*lsmeans can_access / pdiff exp cl ilink;
*lsmeans p_race0 / pdiff exp cl ilink;
by _imputation_;
ods output GEEEmpPEst=param_estimates CovB=param_cov lsmeans=lsm_out_establish
diffs= can_legal0; run;
ods trace off;
*each LSMEANS comparison must be done one at a time ;
```

Step 2: Pool estimates across imputations

```
*pooling estimates across imputations – enforcing rules;
PROC MIANALYZE parms(classvar=level)=param_estimates;
class can_legal0 c_sex0 p_race0 can_access par_edu0 com_inc0;
MODELEFFECTS INTERCEPT can_legal0 can_access par_mon p_race0 par_edu0
com_inc0 c_sex0 timepoint;
run;
```

Step 3: After running LSMEANS statement per predictor (can_legal0, can_access, p_race0), pool comparisons of different levels of predictors.

```
*cannabis legal status----------;
*creating dummy comparison variable to allow for comparing different levels of legal status;
data can_legal0_d;
    set can_legal0_d;
    comparison = cats("Group", can_legal0, "_vs_", _can_legal0);
run;
*double checking name of comparison group was correct;
proc print data=can_legal0_d (obs=10); run;
*ensuring each comparison uses multiple imputations and sorting by ascending order;
proc sort data=can_legal0_d; by comparison _imputation_; run;

proc freq data=can_legal0_d noprint;
    tables comparison / out=comp_counts;
run;

data keep_comparisons;
    set comp_counts;
    if count >= 2;
run;

proc sql;
    create table can_legal0_d_filtered as
    select a.*
    from can_legal0_d a
    inner join keep_comparisons b
    on a.comparison = b.comparison;
quit;

*sorting by ascending order of comparison variable;
```

```
proc sort data=can_legal0_d_filtered; by comparison; run;

*pooling comparisons of different levels of legal status;
proc mianalyze data=can_legal0_d_filtered;
   by comparison;
   modeleffects Estimate;
   stderr StdErr;
run;

*cannabis access----------;
*creating dummy comparison variable to allow for comaparing different levels of cannabis access;
data can_access_d;
   set can_access_d;
   comparison = cats("Group", can_access, "_vs_", _can_access);
run;
*double checking name of comparison group was correct;
proc print data=can_access_d (obs=10); run;
*ensuring each comparison uses multiple imputations and sorting by ascending order;
proc sort data=can_access_d; by comparison _imputation_; run;

proc freq data=can_access_d noprint;
   tables comparison / out=comp_counts;
run;

data keep_comparisons;
   set comp_counts;
   if count >= 2;
run;
```

```
proc sql;
   create table can_access_d_filtered as
   select a.*
   from can_access_d a
   inner join keep_comparisons b
   on a.comparison = b.comparison;
quit;

*sorting by ascending order of comparison variable;
proc sort data=can_access_d_filtered; by comparison; run;

*pooling comparisons of different levels of cannabis access;
proc mianalyze data=can_access_d_filtered;
   by comparison;
   modeleffects Estimate;
   stderr StdErr;
run;

*parent race----------;
*creating dummy comparison variable to allow for comaparing different levels of race;
data p_race0_d;
   set p_race0_d;
   comparison = cats("Group", p_race0, "_vs_", _p_race0);
run;
*double checking name of comparison group was correct;
proc print data=p_race0_d (obs=10); run;
*ensuring each comparison uses multiple imputations and sorting by ascending order;
proc sort data=p_race0_d; by comparison _imputation_; run;
```

```
proc freq data=p_race0_d noprint;
   tables comparison / out=comp_counts;
run;

data keep_comparisons;
   set comp_counts;
   if count >= 2;
run;

proc sql;
   create table p_race0_d_filtered as
   select a.*
   from p_race0_d a
   inner join keep_comparisons b
   on a.comparison = b.comparison;
quit;

*sorting by ascending order of comparison variable;
proc sort data=p_race0_d_filtered; by comparison; run;

*pooling comparisons of different levels of race;
proc mianalyze data=p_race0_d_filtered;
   by comparison;
   modeleffects Estimate;
   stderr StdErr;
run;
```