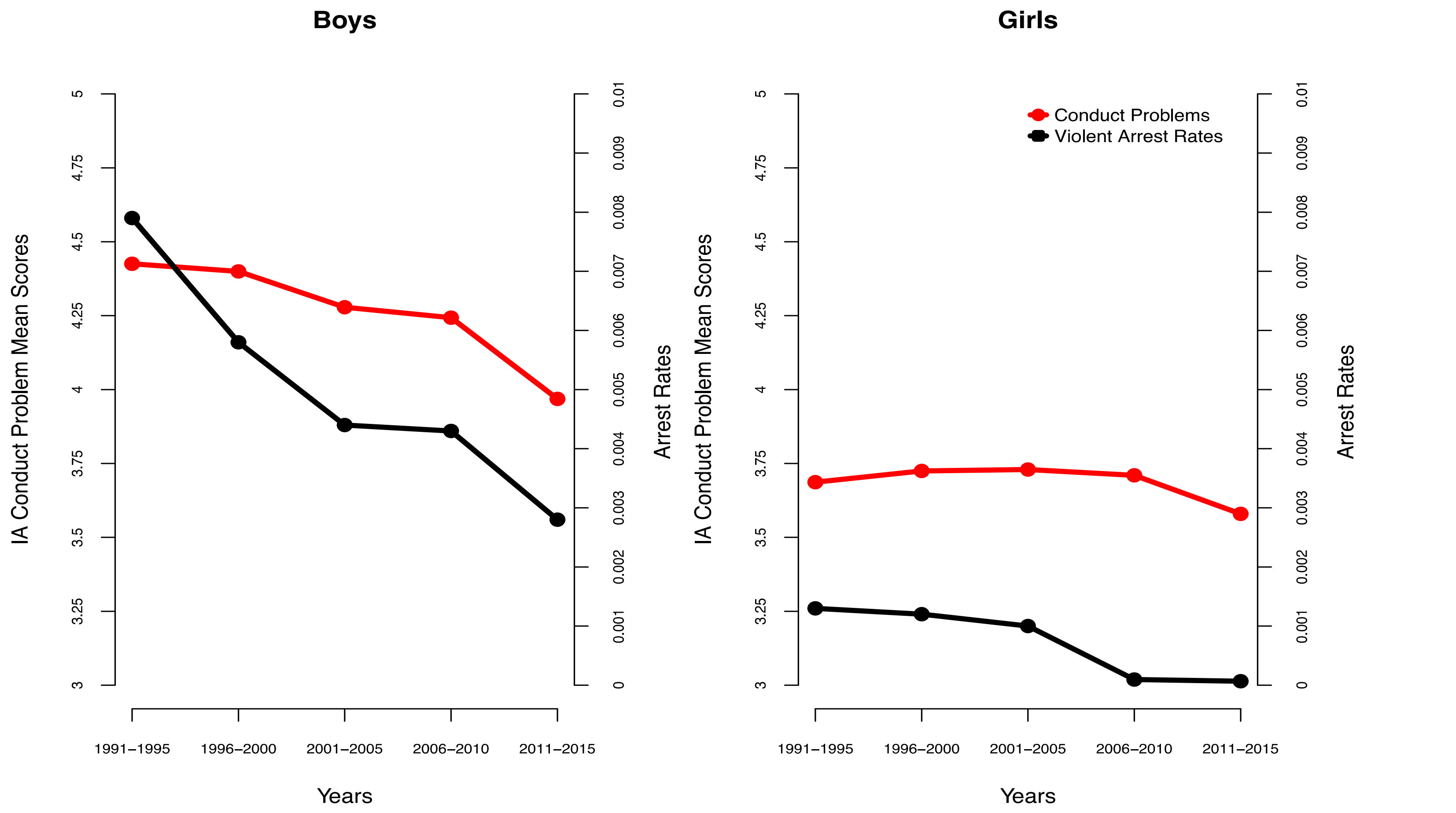
Web Table 1. Unadjusted linear regressions of conduct scores (overall, interpersonal aggression, theft and property damage) on the outcome evenings out among adolescents

	Model 1	**Model 2**	**Model 3**
Beta (SE)	**Overall Score**	**Interpersonal Aggression Score**	**Theft and Property Damage Score**
Evenings out	0.59 (<0.01)	0.25 (<0.01)	0.33 (<0.01)

Web Table 2. Mean and standard error of conduct problems score by year and sex

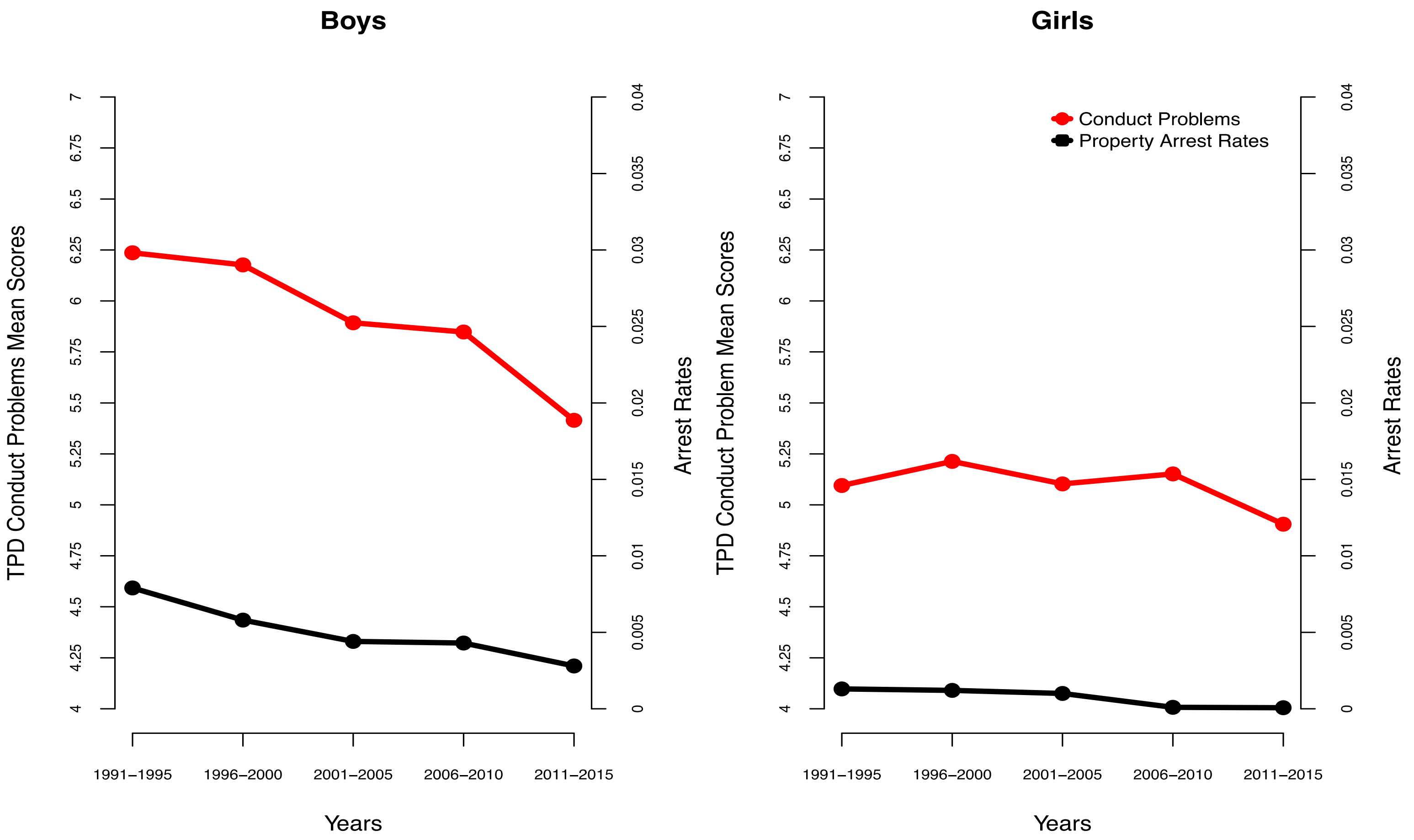
	Boys		**Girls**	
Year	Mean	SE	Mean	SE
1991	10.65	0.05	8.70	0.03
1992	10.73	0.06	8.77	0.03
1993	10.69	0.06	8.78	0.03
1994	10.66	0.06	8.78	0.03
1995	10.59	0.06	8.87	0.03
1996	10.71	0.06	8.98	0.04
1997	10.63	0.07	8.97	0.05
1998	10.48	0.07	8.89	0.04
1999	10.58	0.07	8.98	0.05
2000	10.40	0.07	8.85	0.04
2001	10.45	0.08	8.76	0.04
2002	10.09	0.07	8.85	0.05
2003	10.17	0.07	8.84	0.04
2004	10.15	0.07	8.83	0.04
2005	10.02	0.07	8.88	0.04
2006	10.14	0.07	8.99	0.05
2007	10.05	0.07	8.83	0.04
2008	10.21	0.07	8.89	0.04
2009	10.06	0.07	8.84	0.04
2010	10.00	0.07	8.76	0.04
2011	9.68	0.06	8.65	0.04
2012	9.41	0.06	8.49	0.04
2013	9.42	0.06	8.46	0.04
2014	9.25	0.06	8.44	0.04
2015	9.12	0.06	8.37	0.04

Web Figure 1. All age mean interpersonal aggression scores and violent crime arrest rates* by time period among US adolescents, 1991-2015



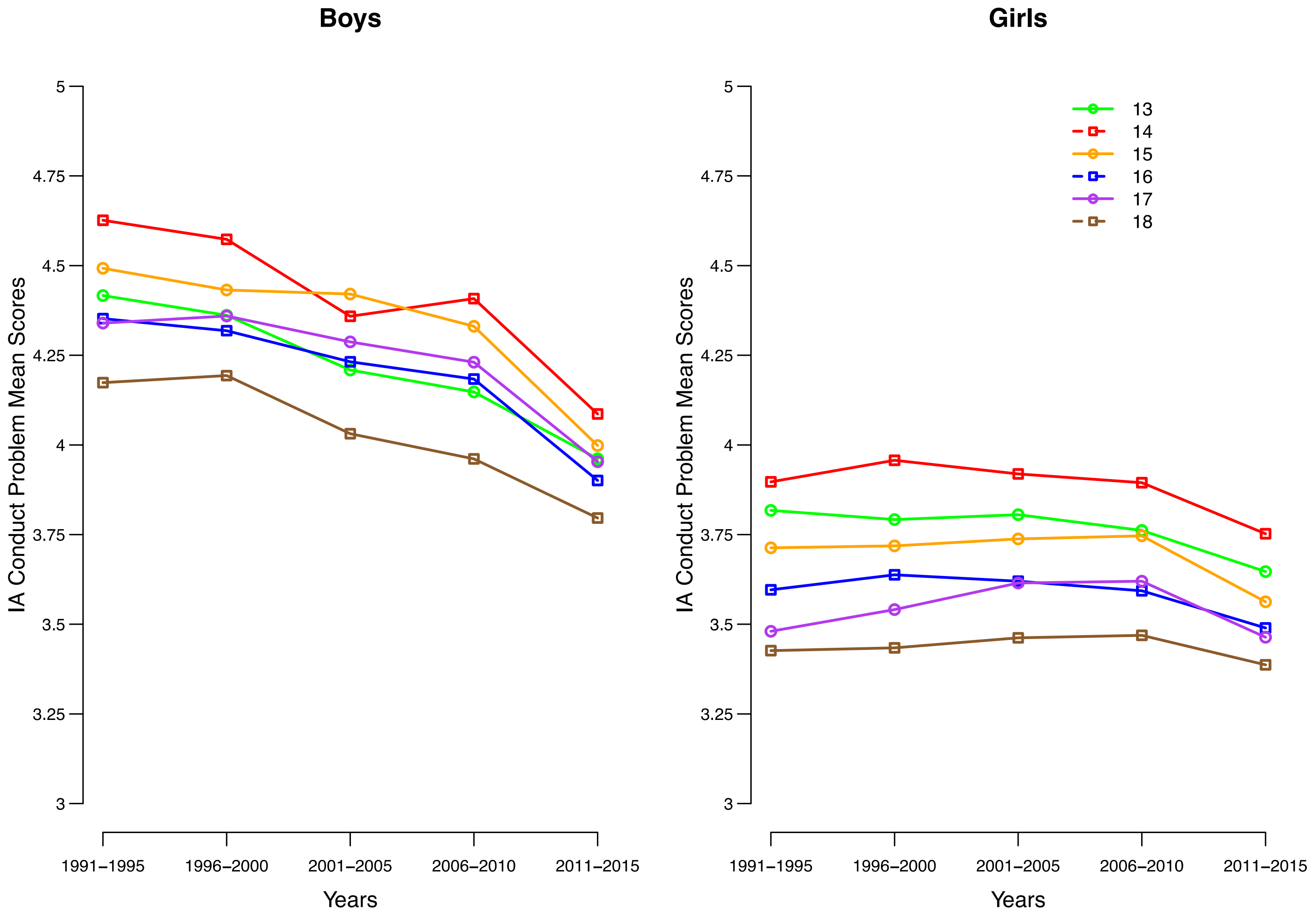
*Arrest rates are of persons ages 10-17 per 100,000 persons ages 10-17 in the resident population from 1991 – 2014. Data drawn from the Office of Juvenile Justice and Delinquency Prevention: http://www.ojjdp.gov/ojstatbb/crime/JAR.asp

Web Figure 2. All age mean theft and property damage scores and property crime arrest rates* by time period among US adolescents, 1991-2015



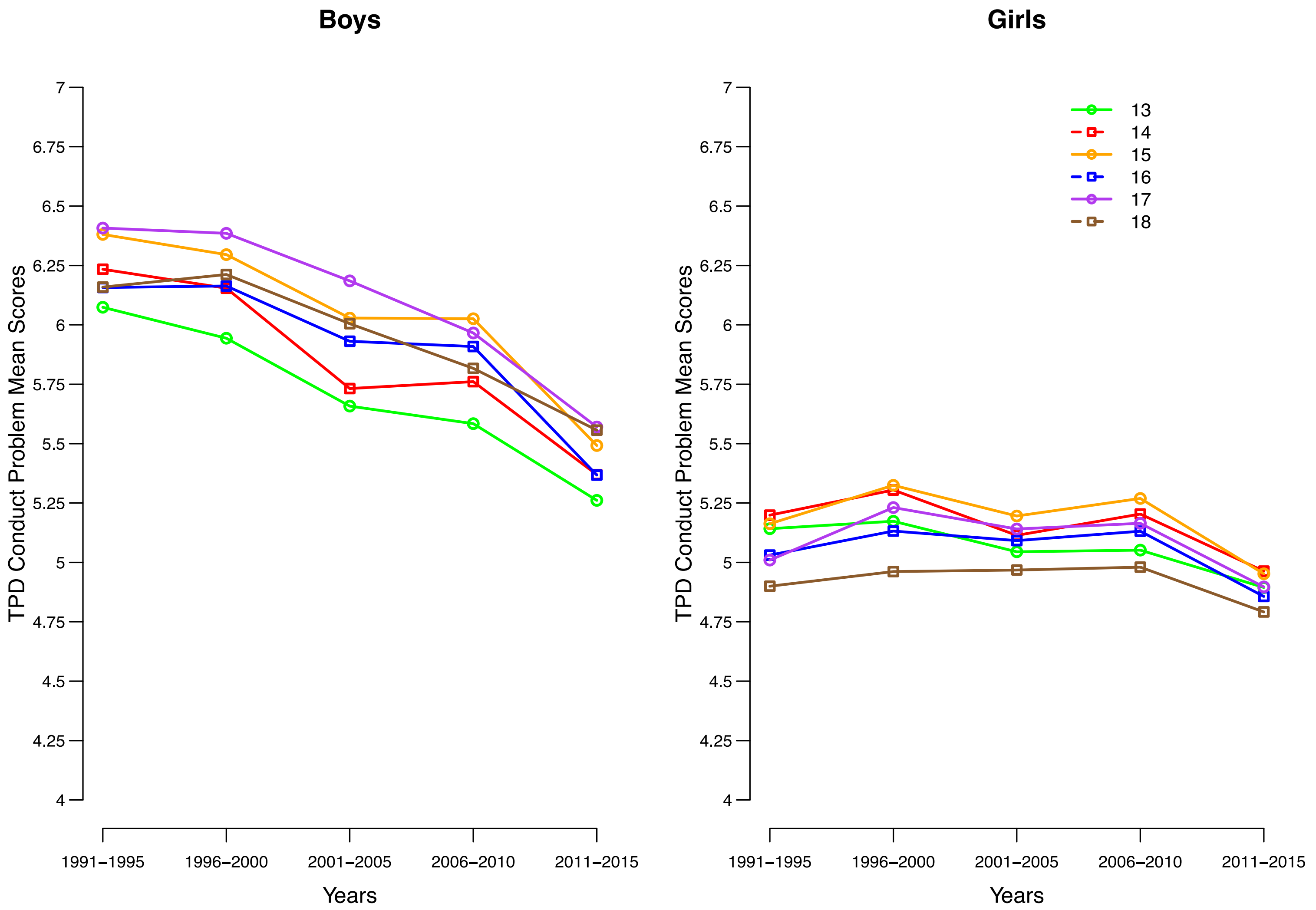
*Arrest rates are of persons ages 10-17 per 100,000 persons ages 10-17 in the resident population from 1991 – 2014. Data drawn from the Office of Juvenile Justice and Delinquency Prevention: http://www.ojjdp.gov/ojstatbb/crime/JAR.asp

Web Figure 3. Mean interpersonal aggression scores by age and time period among US adolescents, 1991-2015



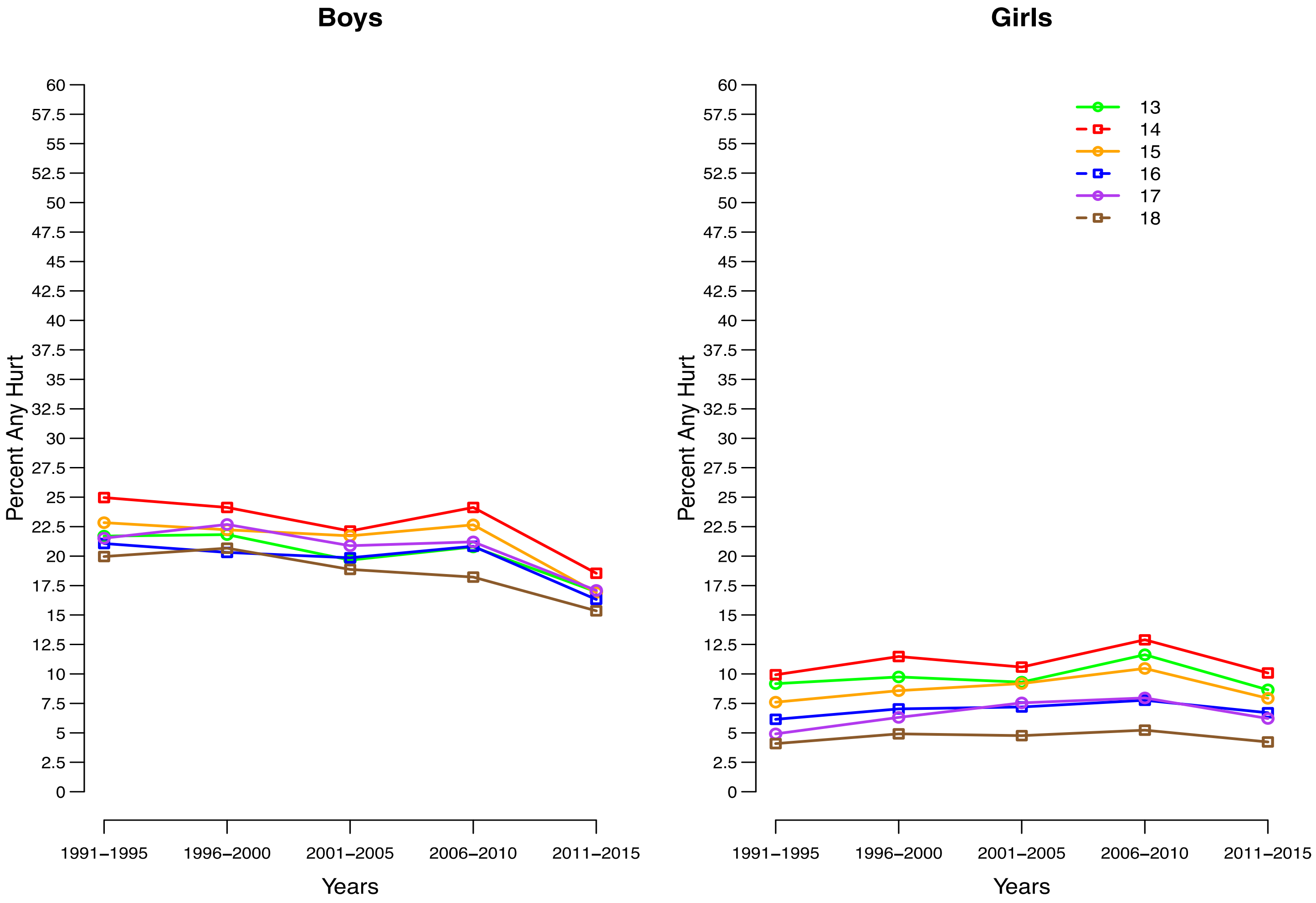
Interpersonal aggression scores are means comprised of respondents answers on a scale of 1 (Never) to 5 (5 or more times) to the three questions: During the last 12 months, how often have you… gotten into a serious fight in school or at work?, taken part in a fight where a group of your friends were against another group?, hurt someone badly enough to need bandages or a doctor?

Web Figure 4. Mean theft and property damage scores by age and time period among US adolescents, 1991-2015

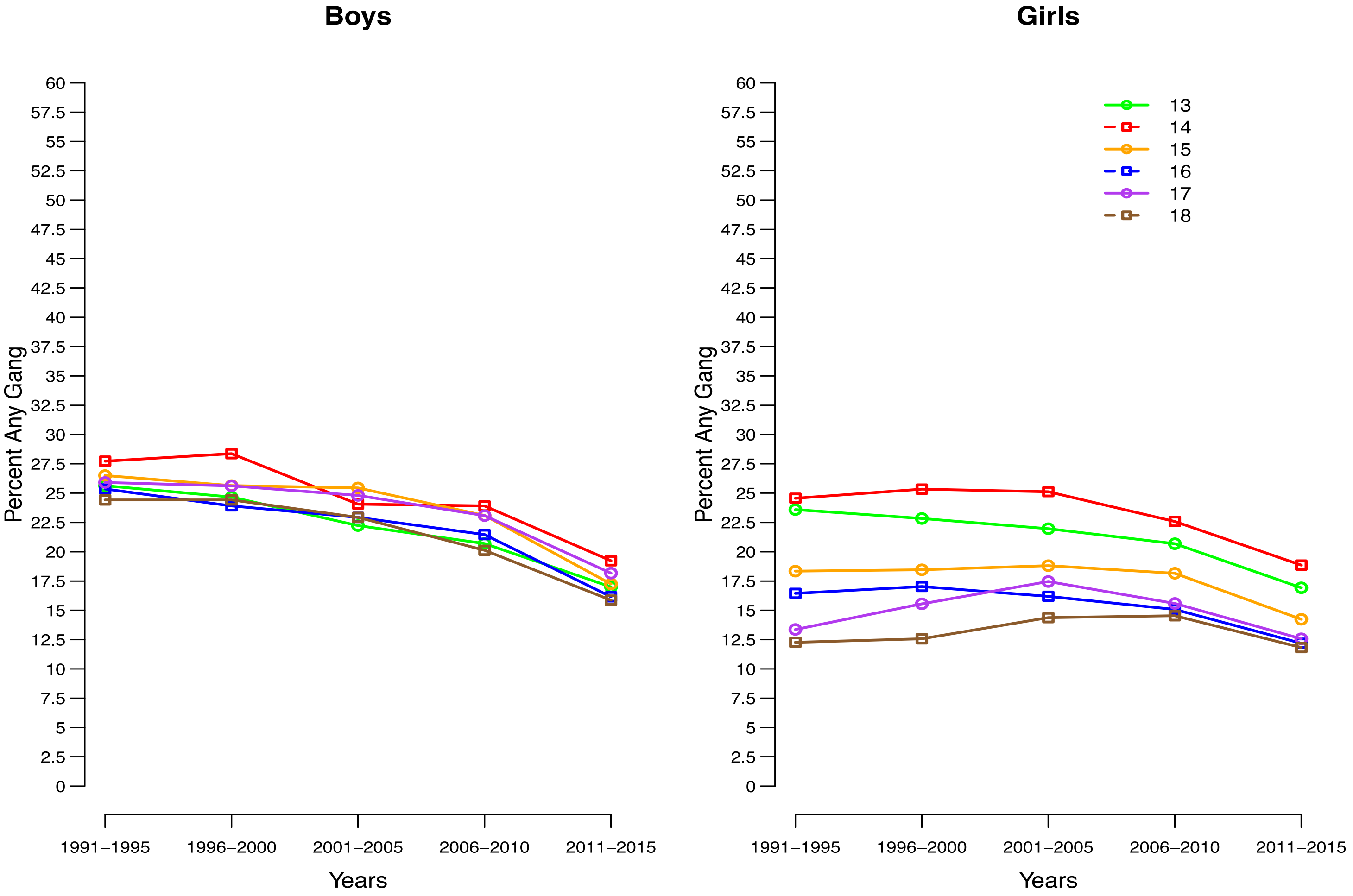


Theft and property damage scores are means comprised of respondents answers on a scale of 1 (Never) to 5 (5 or more times) to the four questions: During the last 12 months, how often have you… taken something not belonging to you worth under $50?, taken something not belonging to you worth over $50?, gone into some house or building when you weren't supposed to be there?, damaged school property on purpose?

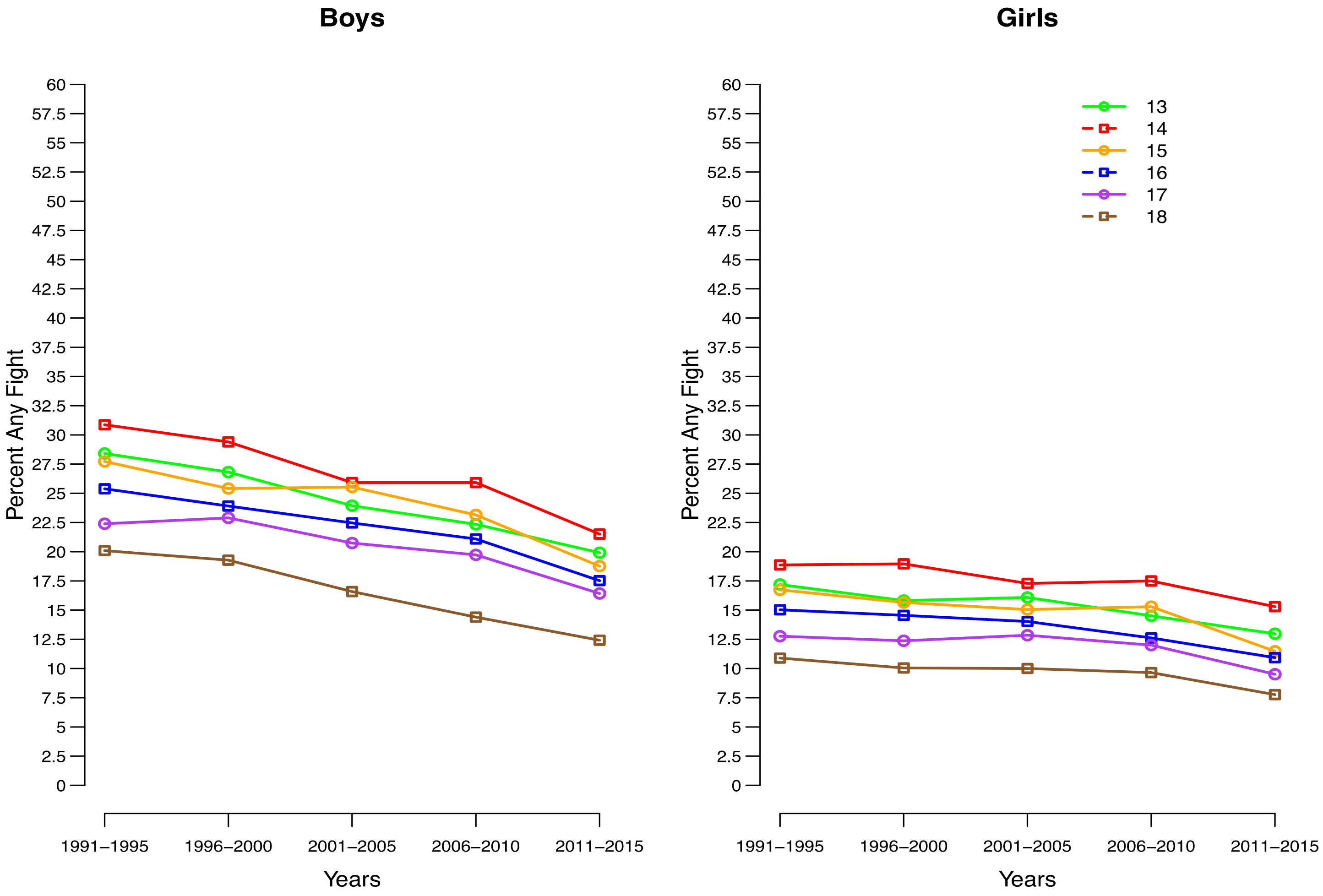
Web Figure 5. Trends over time in hurting someone badly enough to need a doctor by age and time period among US adolescents, 1991-2015



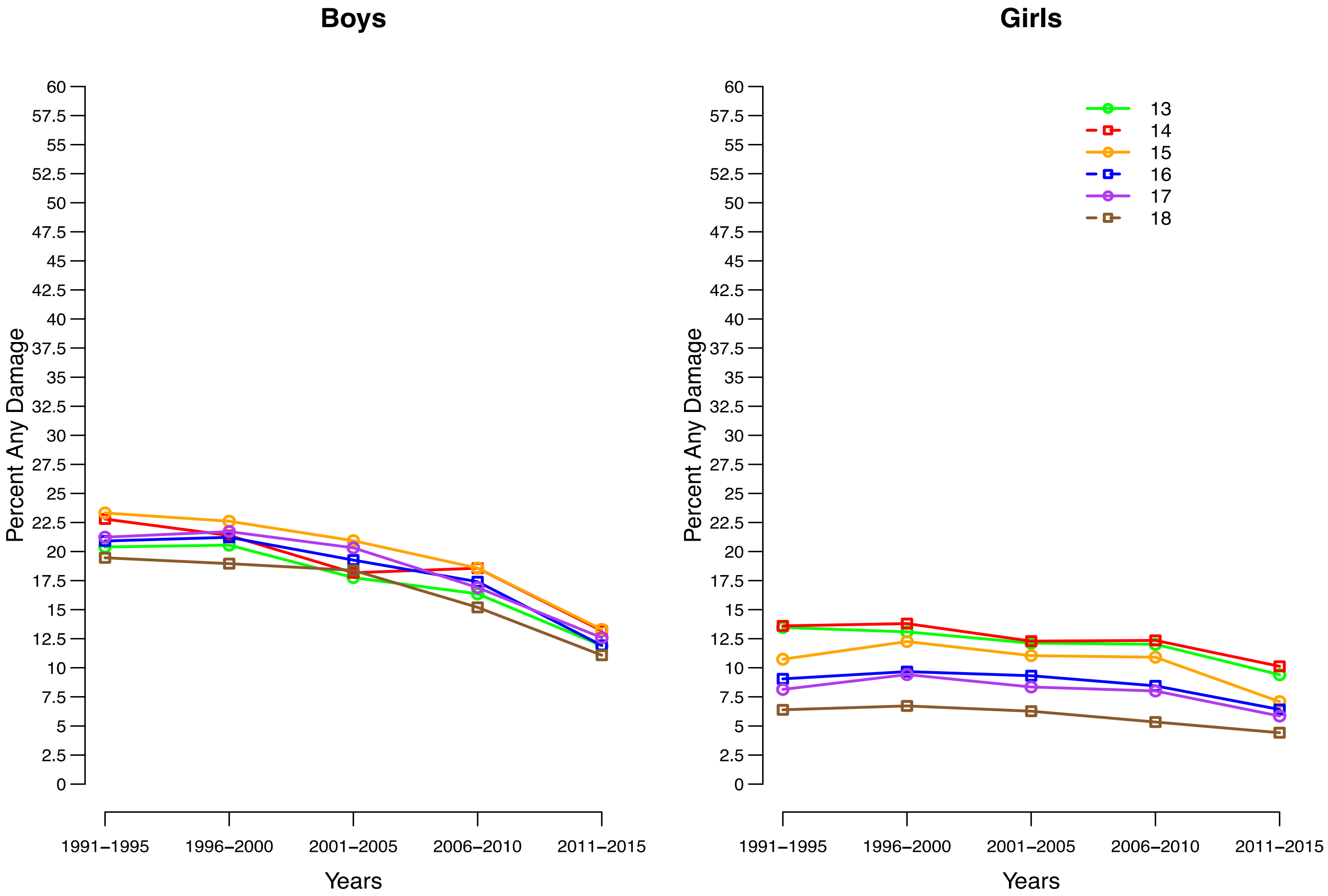
Web Figure 6. Trends over time in taking part in a fight where a group of your friends were against another group by age and time period among US adolescents, 1991-2015



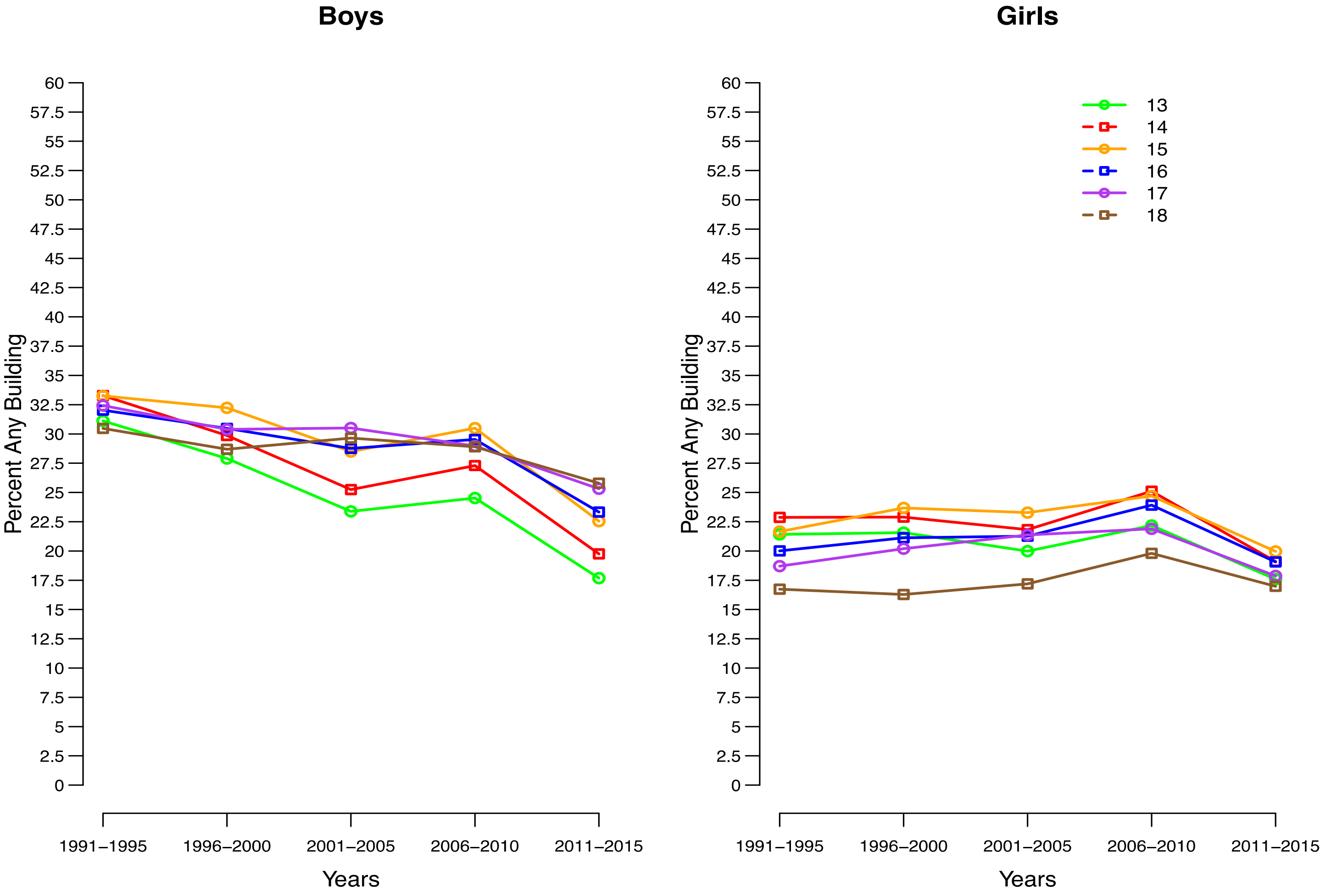
Web Figure 7. Trends over time in getting into a serious fight in school or at work by age and time period among US adolescents, 1991-2015



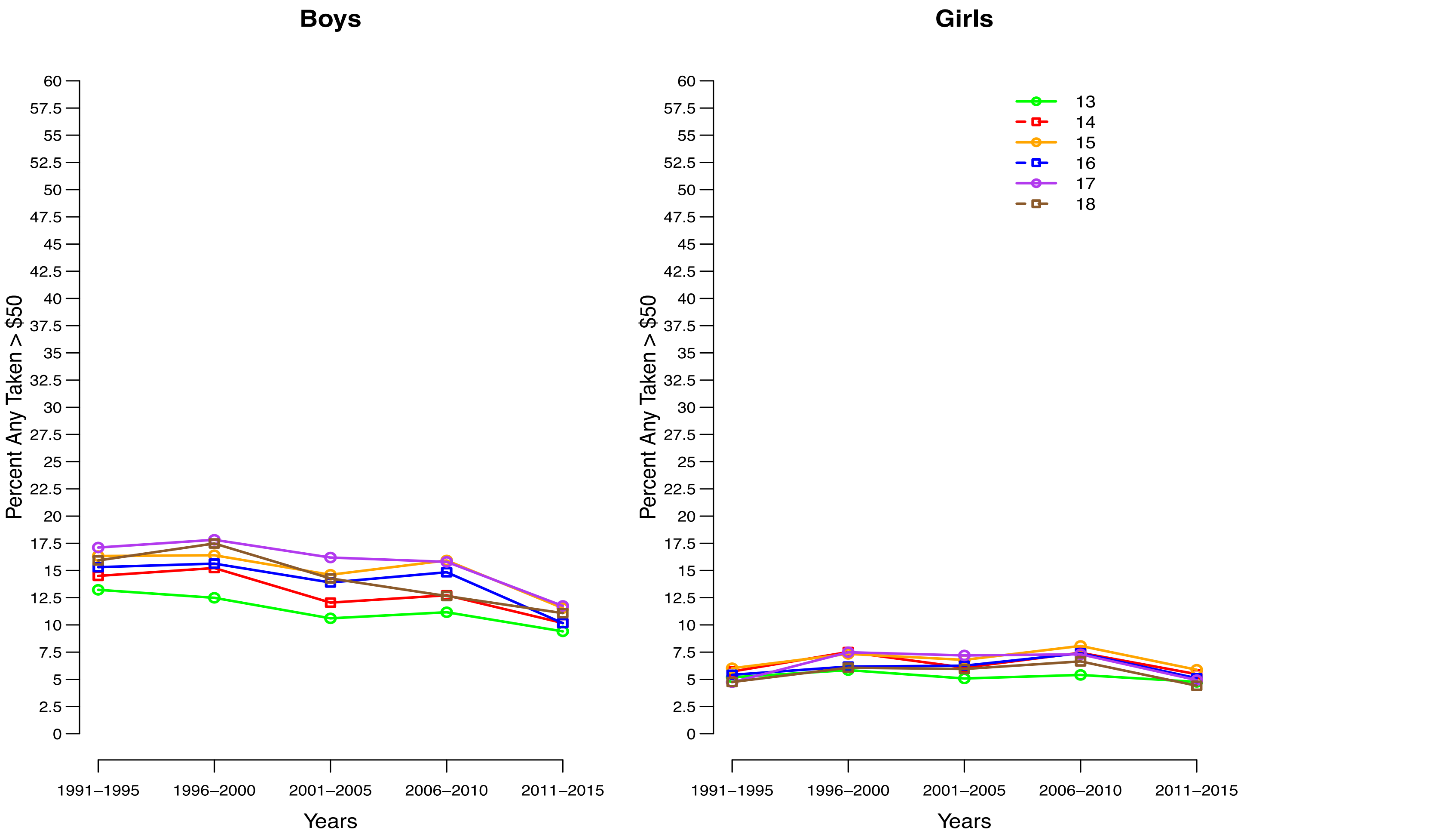
Web Figure 8. Trends over time in damaged school property on purpose by age and time period among US adolescents, 1991-2015



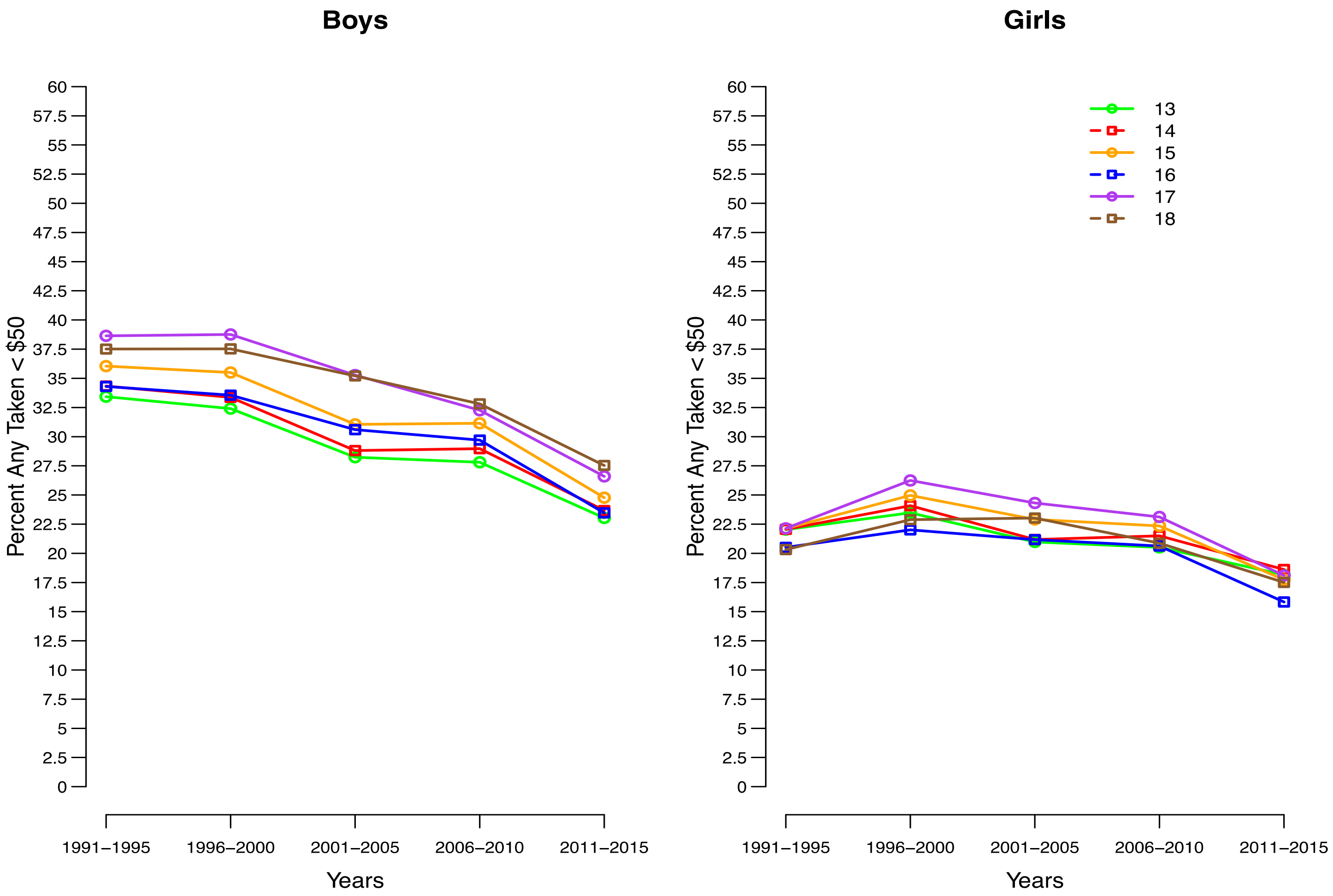
Web Figure 9. Trends over time in entering a house or building when not supposed to by age and time period among US adolescents, 1991-2015



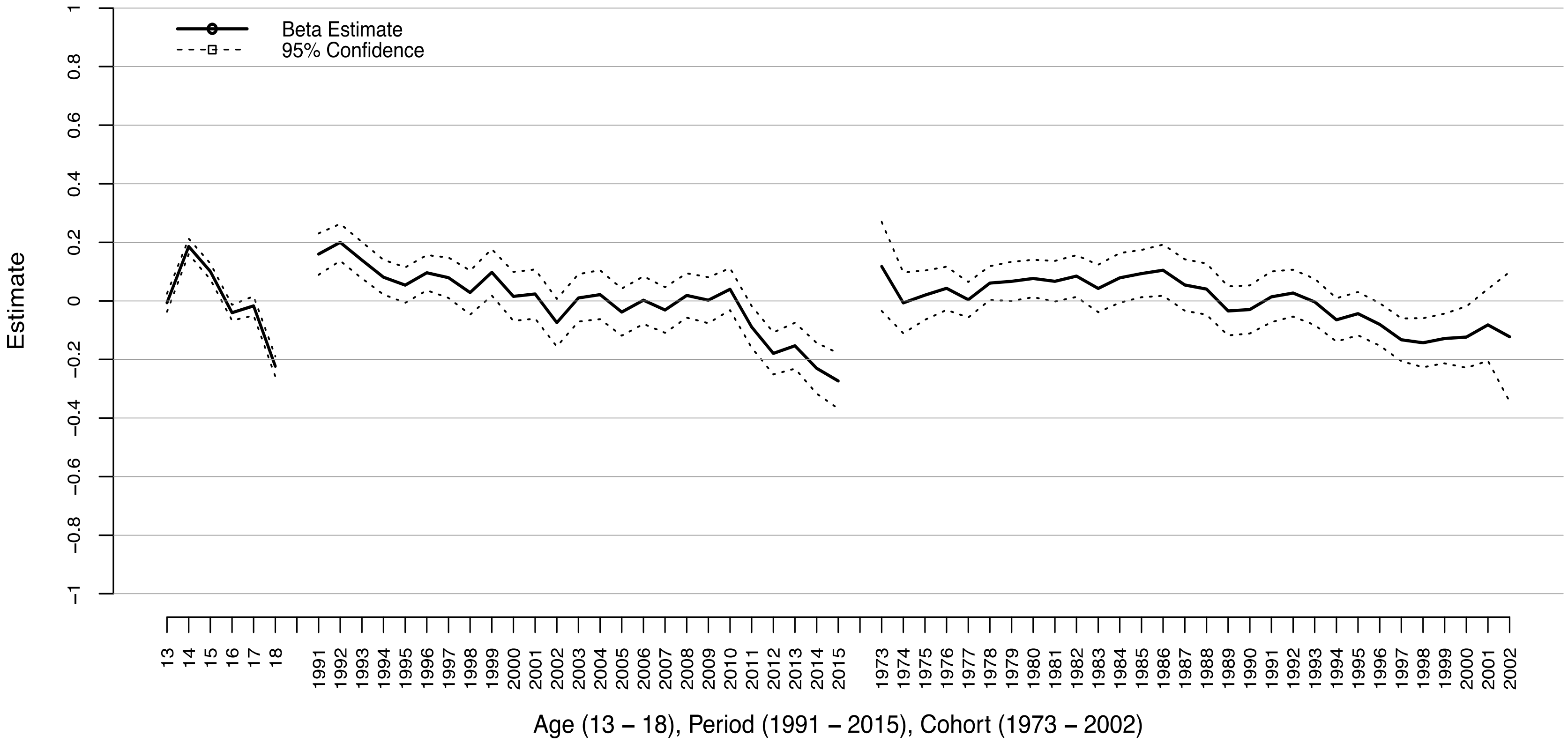
Web Figure 10. Trends over time in taking something not belonging to you worth over $50 by age and time period among US adolescents, 1991-2015



Web Figure 11. Trends over time in taking something not belonging to you worth under $50 by age and time period among US adolescents, 1991-2015

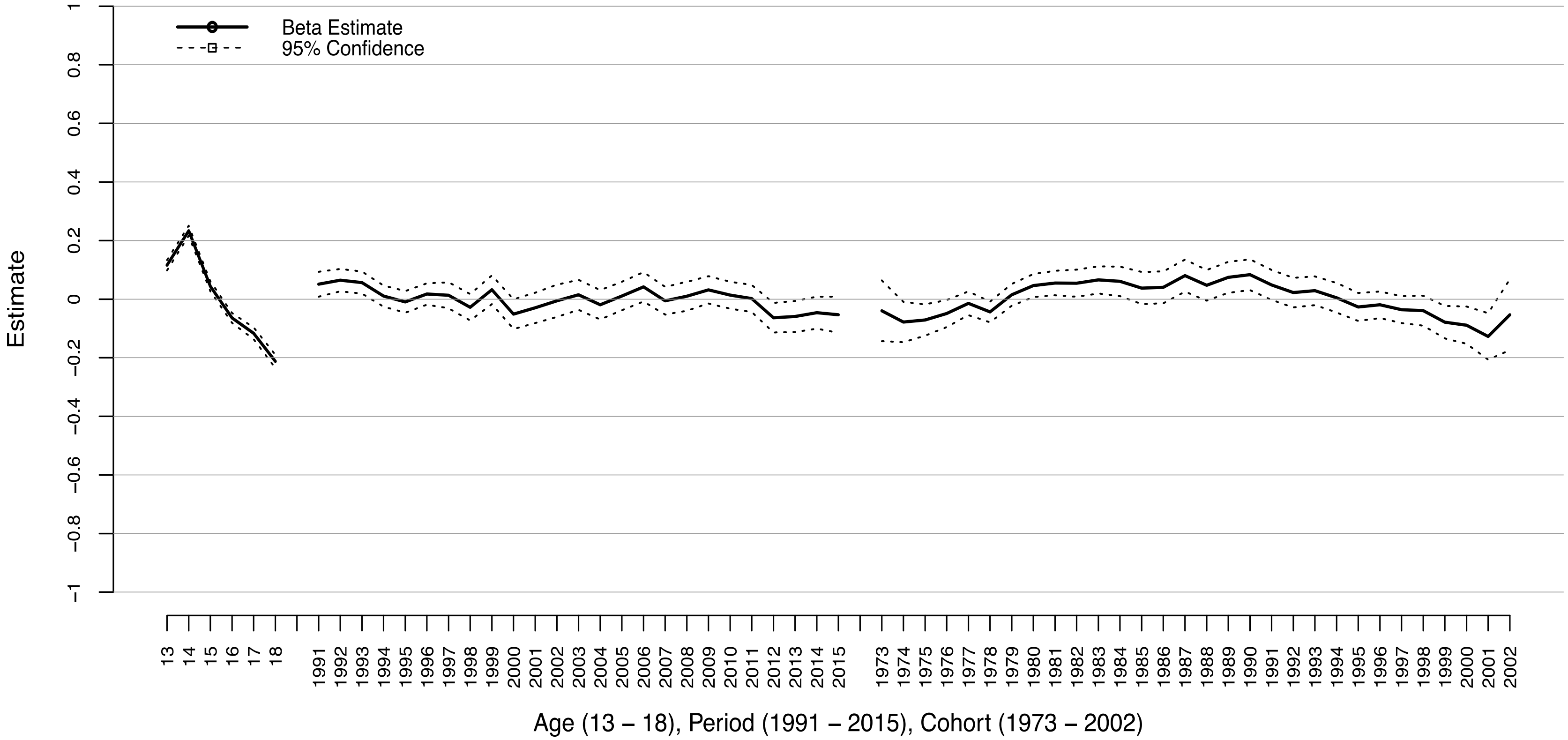


Web Figure 12. Age, period, and cohort effects in interpersonal aggression scores among adolescent boys in the US, 1991-2015



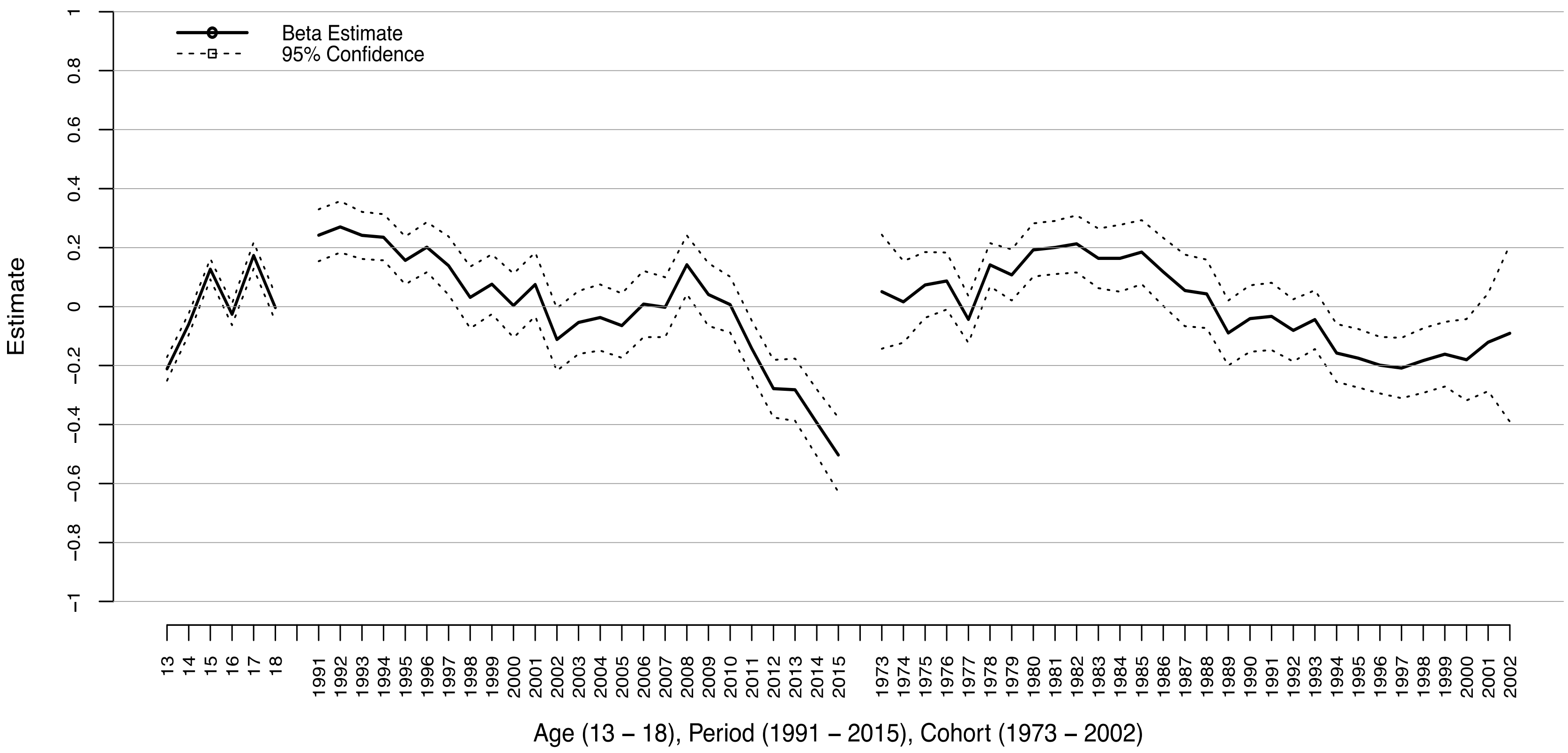
*Each estimate from the Intrinsic Estimator is compared to the mean estimate from the whole sample. For example, those observed in 2015 had an estimate of (-0.273). That estimate indicates that the period effect is significantly lower than the period effect for the whole sample, controlling for age and cohort effects. The gray lines indicate the 95% confidence intervals for the IE estimates.

Web Figure 13. Age, period, and cohort effects in interpersonal aggression scores among adolescent girls in the US, 1991-2015



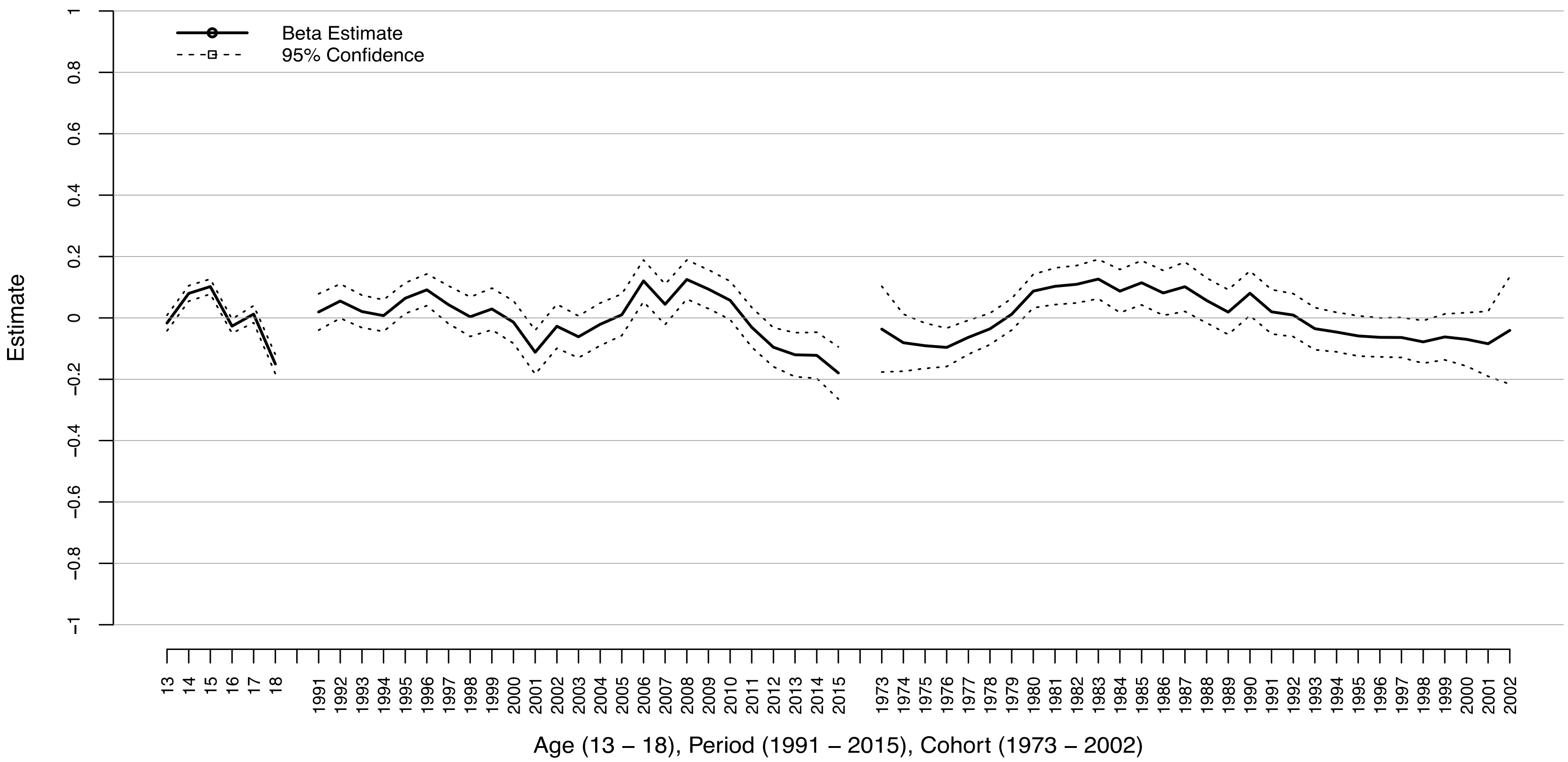
*Each estimate from the Intrinsic Estimator is compared to the mean estimate from the whole sample. For example, those observed in 2013 had an estimate of (-0.059). That estimate indicates that the period effect is not significantly lower than the period effect for the whole sample, controlling for age and cohort effects. The gray lines indicate the 95% confidence intervals for the IE estimates.

Web Figure 14. Age, period, and cohort effects in theft and property damage scores among adolescent boys in the US, 1991-2015



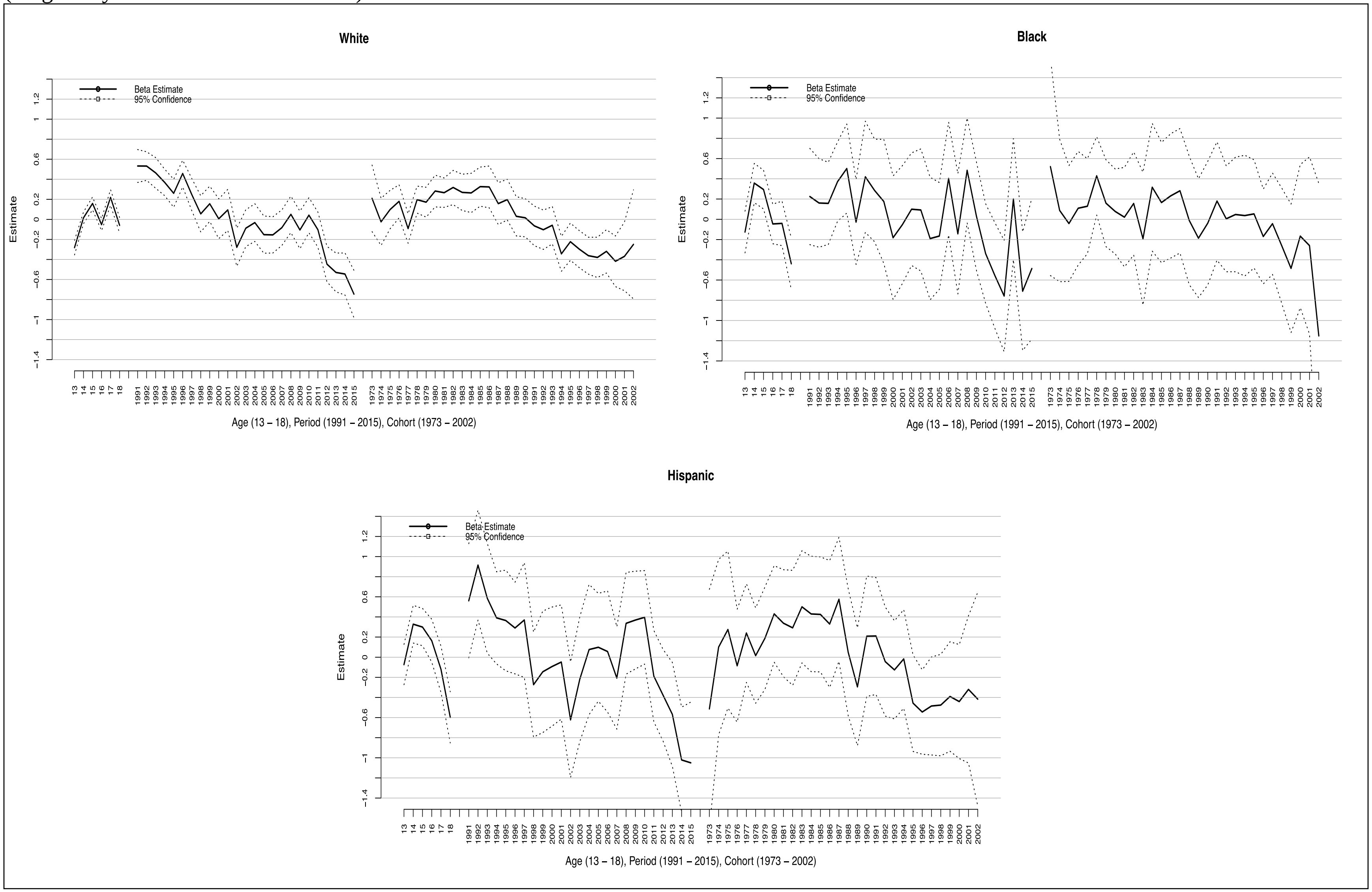
*Each estimate from the Intrinsic Estimator is compared to the mean estimate from the whole sample. For example, those observed in 2015 had an estimate of (-0.503). That estimate indicates that the period effect is significantly lower than the period effect for the whole sample, controlling for age and cohort effects. The gray lines indicate the 95% confidence intervals for the IE estimates.

Web Figure 15. Age, period, and cohort effects in theft and property damage scores among adolescent girls in the US, 1991-2015

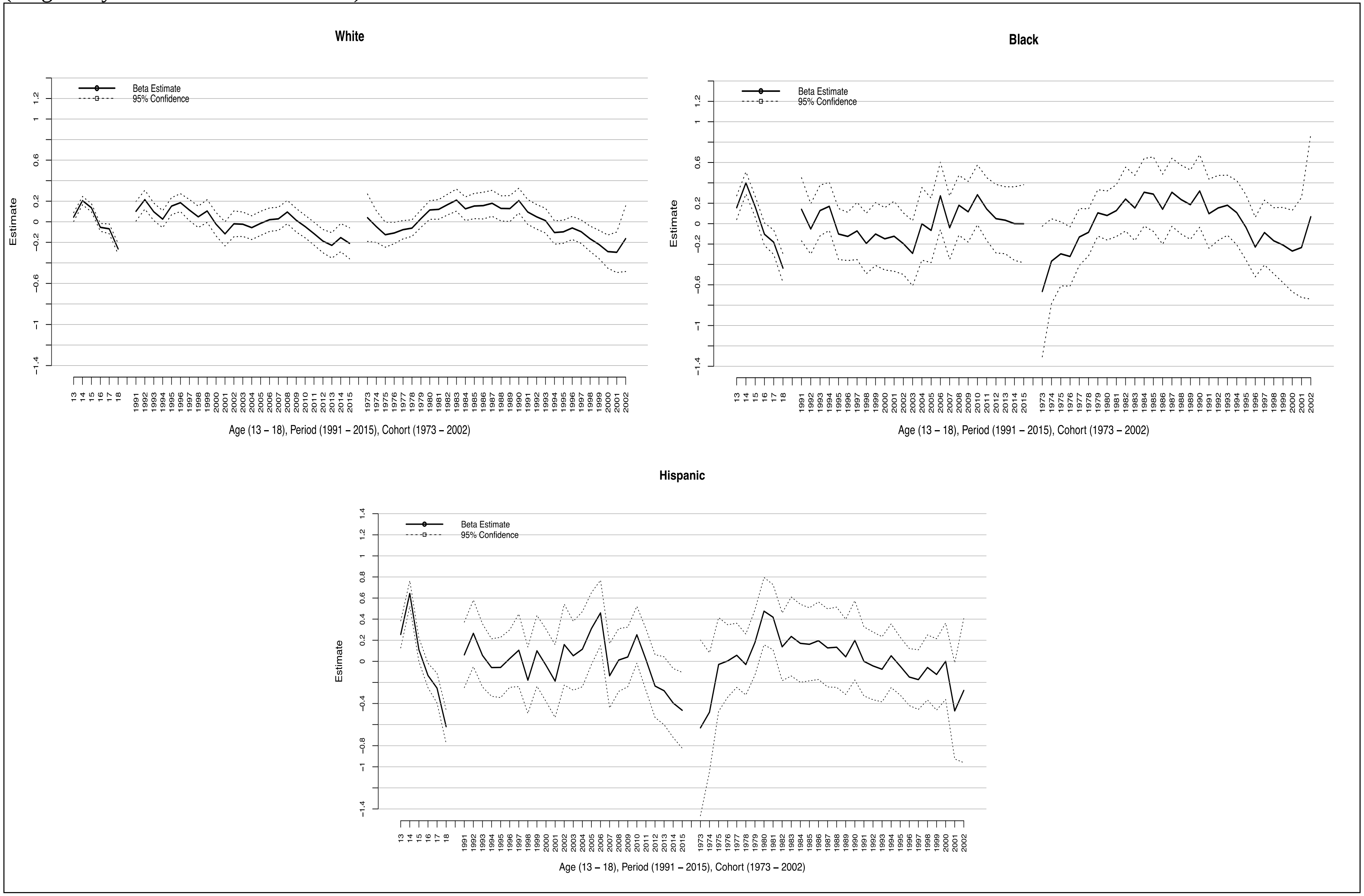


*Each estimate from the Intrinsic Estimator is compared to the mean estimate from the whole sample. For example, those observed in 2015 had an estimate of (-0.179). That estimate indicates that the period effect is significantly lower than the period effect for the whole sample, controlling for age and cohort effects. The gray lines indicate the 95% confidence intervals for the IE estimates.

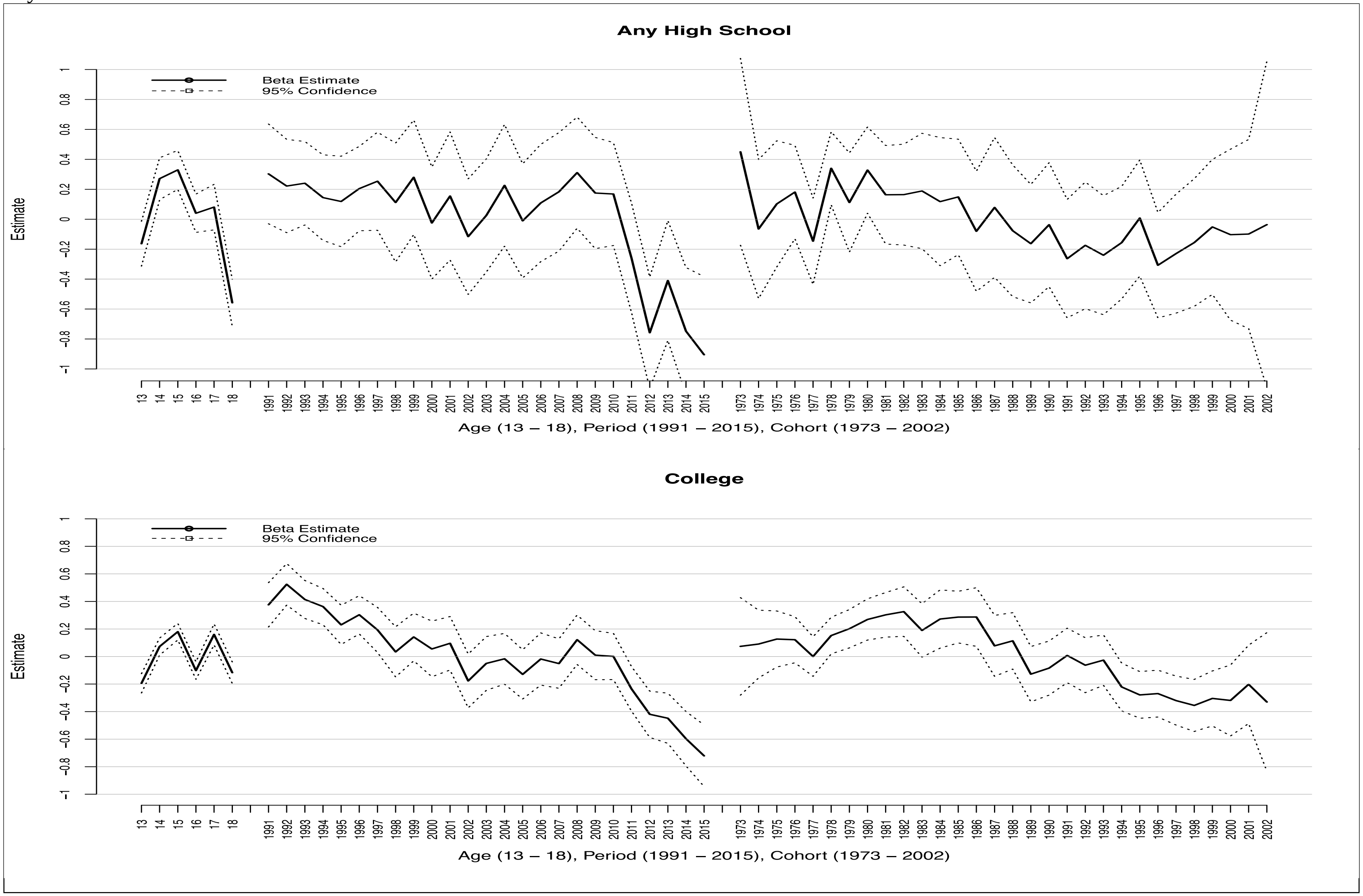
Web Figure 16. Age, period, and cohort effects in overall conduct problem scores among adolescents in the US, 1991-2015, by race, among boys (range for y axis between -1.4 and 1.4)



Web Figure 17. Age, period, and cohort effects in overall conduct problem scores among adolescents in the US, 1991-2015, by race, among girls (range for y axis between -1.4 and 1.4)



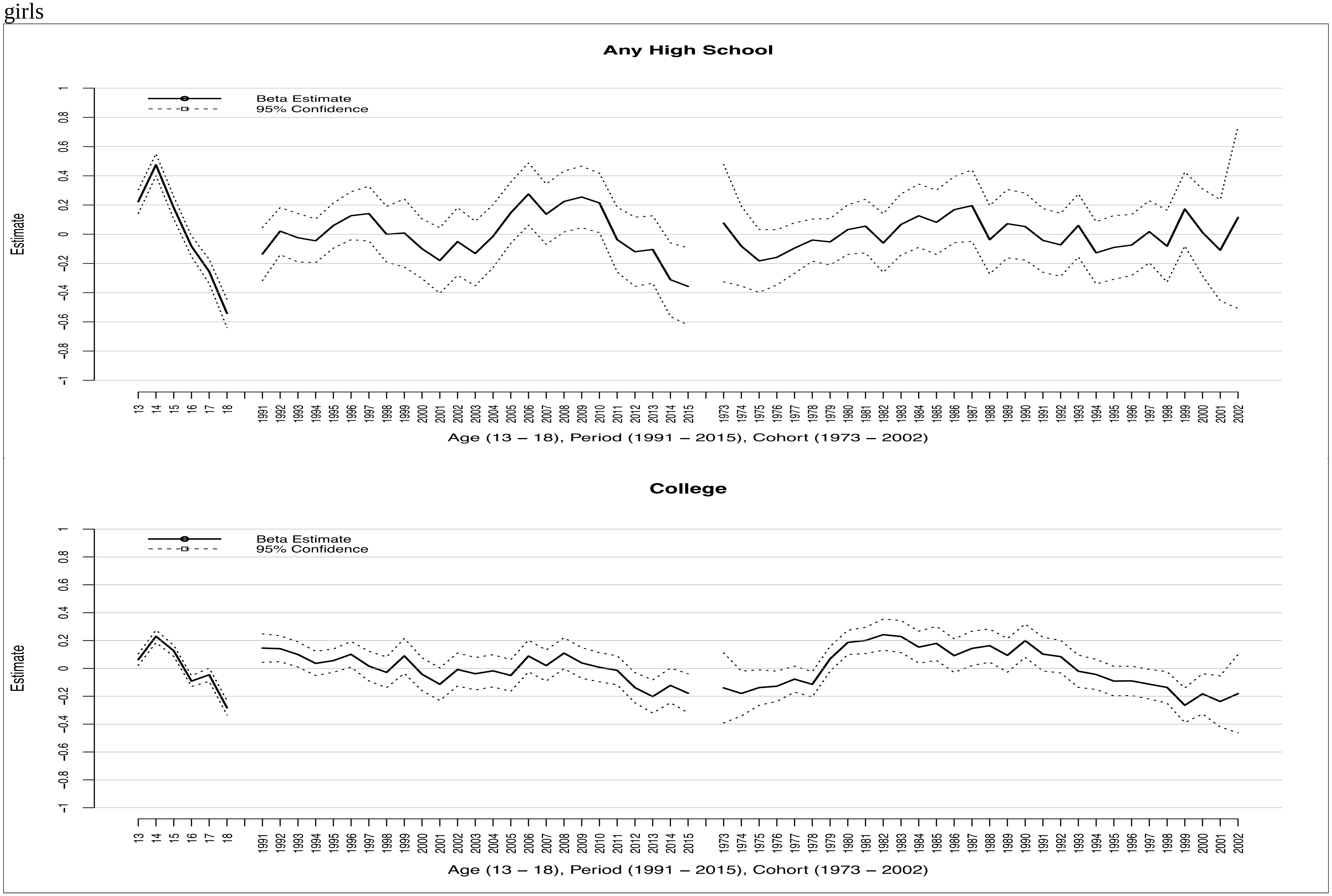
Web Figure 18. Age, period, and cohort effects in overall conduct problem scores among adolescents in the US, 1991-2015, by education, among boys



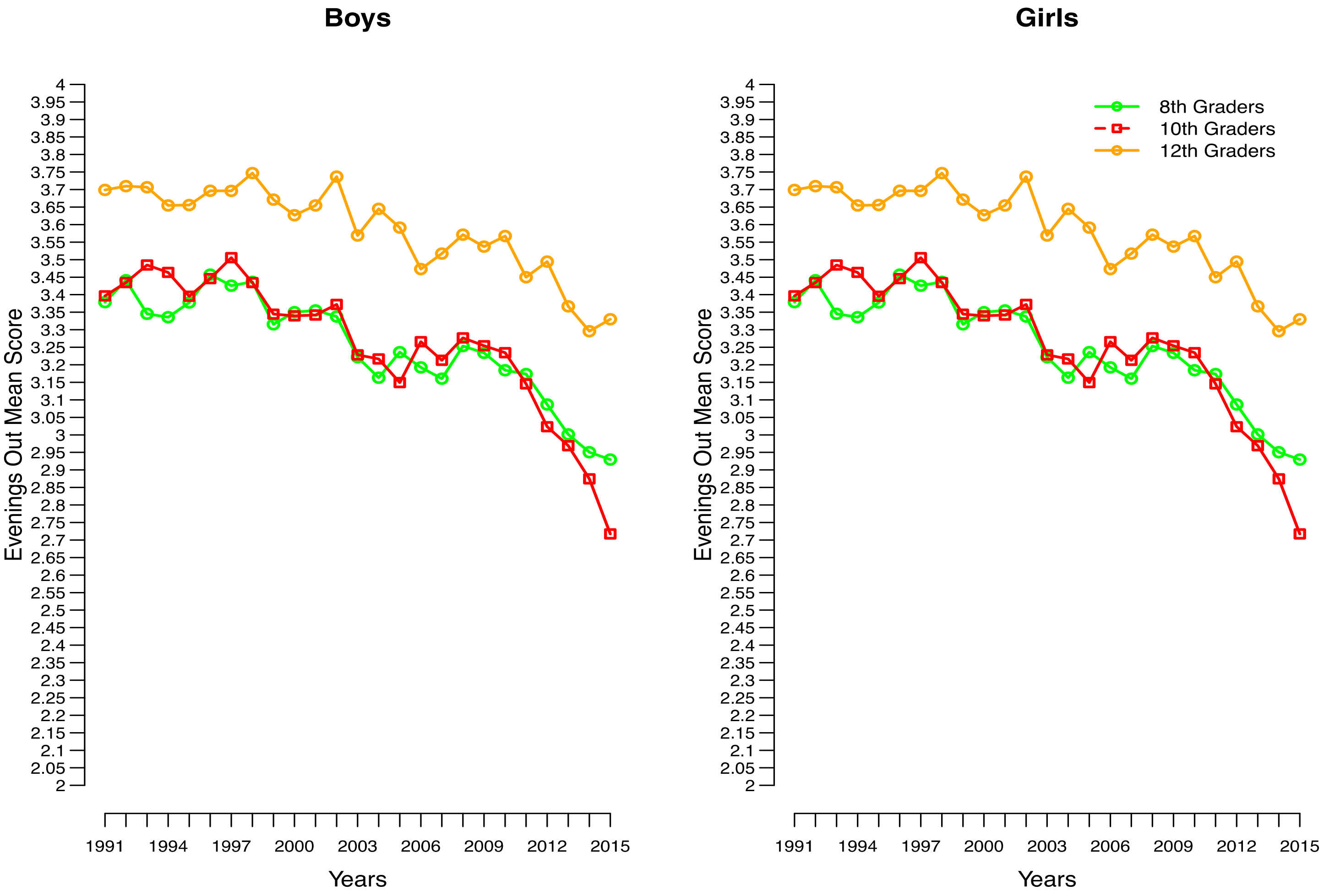
Any High School

College

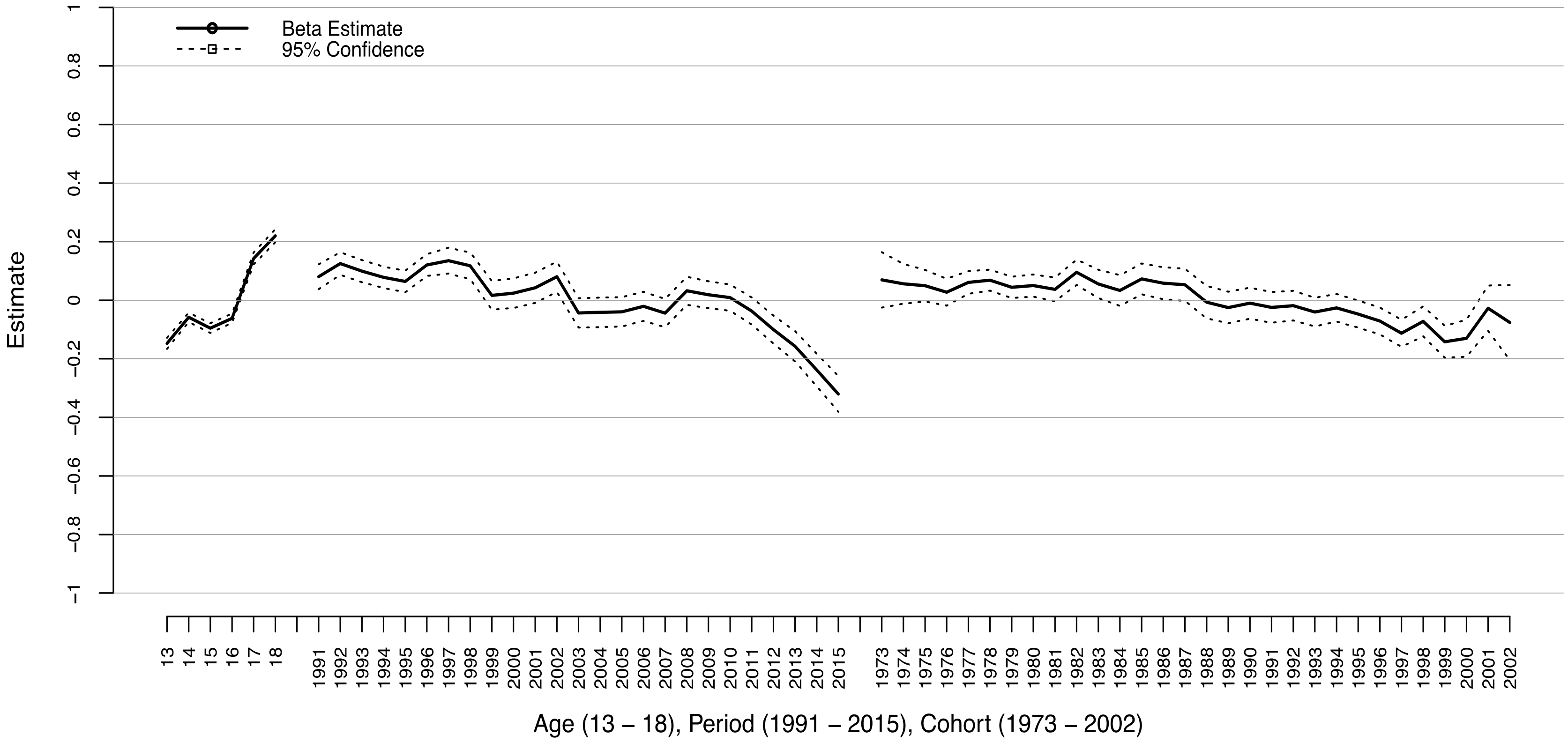
Web Figure 19. Age, period, and cohort effects in overall conduct problem scores among adolescents in the US, 1991-2015, by education, among girls



Web Figure 20. Yearly trends over time in mean reported evenings out per week, among US adolescents, 1991-2015

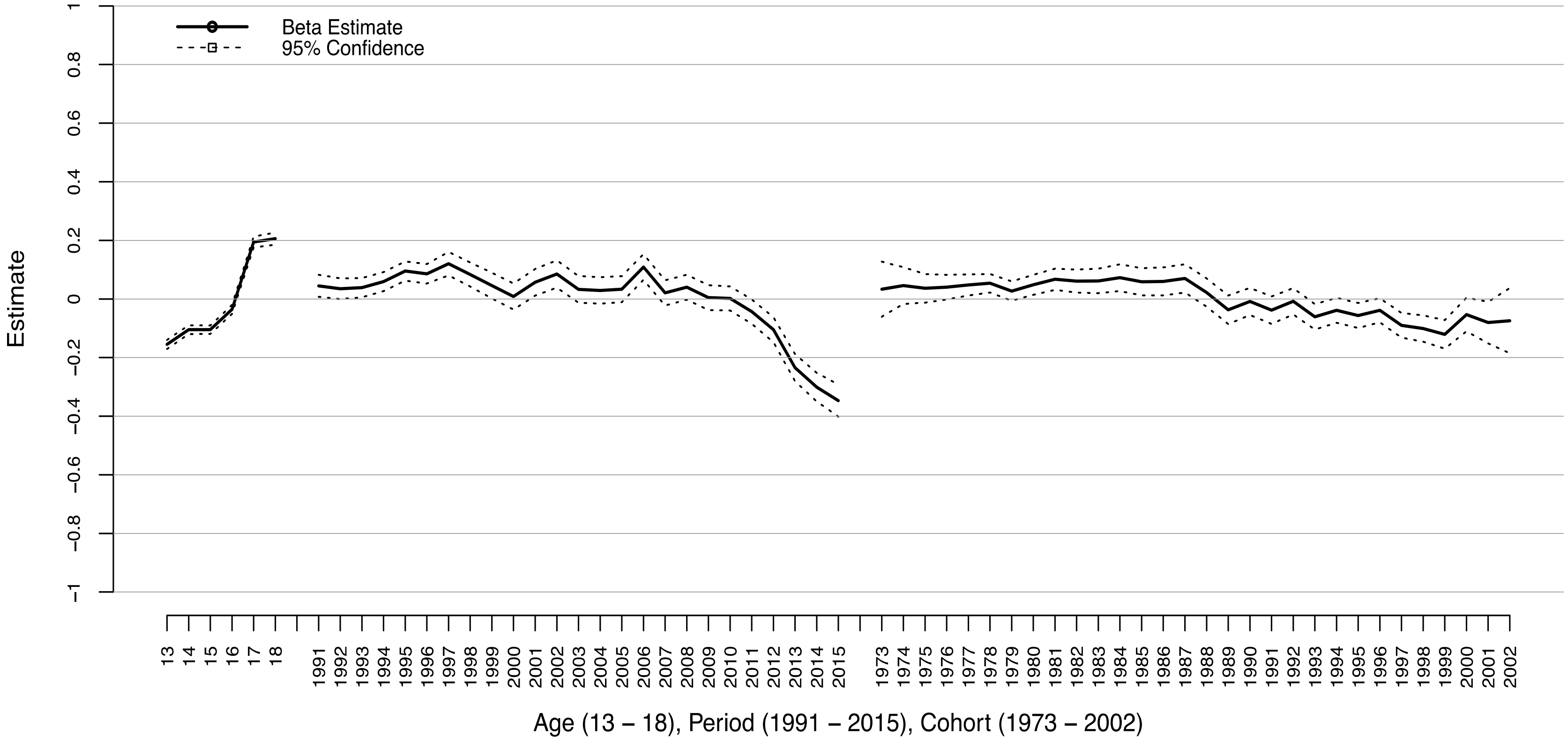


Web Figure 21. Age, period, and cohort effects for evenings out per week among adolescent boys in the US, 1991-2015



*Each estimate from the Intrinsic Estimator is compared to the mean estimate from the whole sample. For example, those observed in 2015 had an estimate of (-0.320). That estimate indicates that the period effect is significantly lower than the period effect for the whole sample, controlling for age and cohort effects. The gray lines indicate the 95% confidence intervals for the IE estimates.

Web Figure 22. Age, period, and cohort effects for evenings out per week among adolescent girls in the US, 1991-2015



*Each estimate from the Intrinsic Estimator is compared to the mean estimate from the whole sample. For example, those observed in 2015 had an estimate of (-0.345). That estimate indicates that the period effect is significantly lower than the period effect for the whole sample, controlling for age and cohort effects. The gray lines indicate the 95% confidence intervals for the IE estimates.

Web Appendix 1. SAS and Stata code for execution of study analyses

```
*Pulling variables from MTF dataset - grade, year and form using macros;
***************************************************************************************************************************
***************************************************************************************************************************
***************************************************************************************************************************
                                                                                        *12th GRADE*
***************************************************************************************************************************
***************************************************************************************************************************
***************************************************************************************************************************
;
%macro twelveSetupCore(outfile, infile, year, grade, id, sex, mother_ed, father_ed, evenings);

data work.&outfile;
set twelve.&infile;

year=&year;
grade=&grade;
id=&id;
sex=&sex; *1=male, 2=female;

if &mother_ed in (1,2) then mother_ed = 1; *some highschool or lower;
if &mother_ed = 3 then mother_ed =2; *finished highschool;
if &mother_ed in (4,5,6) then mother_ed = 3; *some college to graduate school;
if &mother_ed = 7 then mother_ed = 0; *don't know;

if &father_ed in (1,2) then father_ed = 1; *some highschool or lower;
if &father_ed = 3 then father_ed =2; *finished highschool;
if &father_ed in (4,5,6) then father_ed = 3; *some college to graduate school;
if &father_ed = 7 then father_ed = 0; *don't know;
evenings= &evenings;*1=<1, 2=one, 3=two, 4=three, 5=four-five, 6=six-seven;

run;

%mend;

%twelveSetupCore(y1991_1, y1991_f1, 1991, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y1992_1, y1992_f1, 1992, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y1993_1, y1993_f1, 1993, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y1994_1, y1994_f1, 1994, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y1995_1, y1995_f1, 1995, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y1996_1, y1996_f1, 1996, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y1997_1, y1997_f1, 1997, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y1998_1, y1998_f1, 1998, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y1999_1, y1999_f1, 1999, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2000_1, y2000_f1, 2000, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2001_1, y2001_f1, 2001, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2002_1, y2002_f1, 2002, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2003_1, y2003_f1, 2003, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2004_1, y2004_f1, 2004, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2005_1, y2005_f1, 2005, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2006_1, y2006_f1, 2006, 12, v4, v150, v164, v163, v194);
```

```
%twelveSetupCore(y2007_1, y2007_f1, 2007, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2008_1, y2008_f1, 2008, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2009_1, y2009_f1, 2009, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2010_1, y2010_f1, 2010, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2011_1, y2011_f1_0324, 2011, 12, v4, v150, v164, v163, v194);
%twelveSetupCore(y2012_1, y2012_core, 2012, 12, RESPONDENT_ID, v2150, v2164, v2163, v2194);
%twelveSetupCore(y2013_1, y2013_f1, 2013, 12, v6, v2150, v2164, v2163, v2194);
%twelveSetupCore(y2014_1, y2014_f1, 2014, 12, RESPONDENT_ID, v2150, v2164, v2163, v2194);
%twelveSetupCore(y2015_1, y2015_f1, 2015, 12, RESPONDENT_ID, v2150, v2164, v2163, v2194);

%macro twelveSetupf2_6(outfile, infile, year, grade, id, form, taken, building, damaged, taken1, fight, gang, hurt);

data work.&outfile;
set twelve.&infile;

year=&year;
grade=&grade;
id=&id;
form=&form;

*CP;
taken=&taken;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
building=&building;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
damaged=&damaged;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
takenl=&taken1;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;

fight=&fight;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
gang=&gang;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
hurt=&hurt; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;

run;

%mend;

%twelveSetupf2_6(y1991_3, y1991_f3, 1991, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y1991_7, y1991_f7, 1991, 12, v4, 6, v6287, v6291, v6293, v6286, v6282, v6283, v6284);
%twelveSetupf2_6(y1992_3, y1992_f3, 1992, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y1992_7, y1992_f7, 1992, 12, v4, 6, v6287, v6291, v6293, v6286, v6282, v6283, v6284);
%twelveSetupf2_6(y1993_3, y1993_f3, 1993, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y1993_7, y1993_f7, 1993, 12, v4, 6, v6287, v6291, v6293, v6286, v6282, v6283, v6284);
%twelveSetupf2_6(y1994_3, y1994_f3, 1994, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y1994_7, y1994_f7, 1994, 12, v4, 6, v6287, v6291, v6293, v6286, v6282, v6283, v6284);
%twelveSetupf2_6(y1995_3, y1995_f3, 1995, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y1995_7, y1995_f7, 1995, 12, v4, 6, v6287, v6291, v6293, v6286, v6282, v6283, v6284);
%twelveSetupf2_6(y1996_3, y1996_f3, 1996, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y1996_7, y1996_f7, 1996, 12, v4, 6, v6287, v6291, v6293, v6286, v6282, v6283, v6284);
%twelveSetupf2_6(y1997_3, y1997_f3, 1997, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y1998_3, y1998_f3, 1998, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y1999_3, y1999_f3, 1999, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2000_3, y2000_f3, 2000, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2001_3, y2001_f3, 2001, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2002_3, y2002_f3, 2002, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2003_3, y2003_f3, 2003, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2004_3, y2004_f3, 2004, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2005_3, y2005_f3, 2005, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
```

```
%twelveSetupf2_6(y2006_3, y2006_f3, 2006, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2007_3, y2007_f3, 2007, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2008_3, y2008_f3, 2008, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2009_3, y2009_f3, 2009, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2010_3, y2010_f3, 2010, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2011_3, y2011_f2, 2011, 12, v4, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2012_3, y2012_f3, 2012, 12, RESPONDENT_ID, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2013_3, y2013_f3, 2013, 12, v6, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2014_3, y2014_f3, 2014, 12, RESPONDENT_ID, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);
%twelveSetupf2_6(y2015_3, y2015_f3, 2015, 12, RESPONDENT_ID, 2, v2286, v2290, v2292, v2285, v2281, v2282, v2283);

*Merging years for 12th graders;
*1991;
proc sort data = y1991_1; by id; run;
proc sort data = y1991_3; by id; run;
proc sort data = y1991_7; by id; run;
data y1991_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y1991_1 y1991_3 y1991_7;
by id;
run;

*1992;
proc sort data = y1992_1; by id; run;
proc sort data = y1992_3; by id; run;
proc sort data = y1992_7; by id; run;
data y1992_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y1992_1 y1992_3 y1992_7;
by id;
run;

*1993;
proc sort data = y1993_1; by id; run;
proc sort data = y1993_3; by id; run;
proc sort data = y1993_7; by id; run;
data y1993_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y1993_1 y1993_3 y1993_7;
by id;
run;

*1994;
proc sort data = y1994_1; by id; run;
proc sort data = y1994_3; by id; run;
proc sort data = y1994_7; by id; run;
data y1994_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y1994_1 y1994_3 y1994_7;
by id;
run;

*1995;
proc sort data = y1995_1; by id; run;
proc sort data = y1995_3; by id; run;
proc sort data = y1995_7; by id; run;
data y1995_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y1995_1 y1995_3 y1995_7;
by id;
```

```
run;

*1996;
proc sort data = y1996_1; by id; run;
proc sort data = y1996_3; by id; run;
proc sort data = y1996_7; by id; run;
data y1996_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y1996_1 y1996_3 y1996_7;
by id;
run;

*1997;
proc sort data = y1997_1; by id; run;
proc sort data = y1997_3; by id; run;
data y1997_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y1997_1 y1997_3;
by id;
run;

*1998;
proc sort data = y1998_1; by id; run;
proc sort data = y1998_3; by id; run;
data y1998_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y1998_1 y1998_3;
by id;
run;

*1999;
proc sort data = y1999_1; by id; run;
proc sort data = y1999_3; by id; run;
data y1999_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y1999_1 y1999_3;
by id;
run;

*2000;
proc sort data = y2000_1; by id; run;
proc sort data = y2000_3; by id; run;
data y2000_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2000_1 y2000_3;
by id;
run;

*2001;
proc sort data = y2001_1; by id; run;
proc sort data = y2001_3; by id; run;
data y2001_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2001_1 y2001_3;
by id;
run;

*2002;
proc sort data = y2002_1; by id; run;
proc sort data = y2002_3; by id; run;
data y2002_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
```

```
merge y2002_1 y2002_3;
by id;
run;

*2003;
proc sort data = y2003_1; by id; run;
proc sort data = y2003_3; by id; run;
data y2003_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2003_1 y2003_3;
by id;
run;

*2004;
proc sort data = y2004_1; by id; run;
proc sort data = y2004_3; by id; run;
data y2004_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2004_1 y2004_3;
by id;
run;

*2005;
proc sort data = y2005_1; by id; run;
proc sort data = y2005_3; by id; run;
data y2005_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2005_1 y2005_3;
by id;
run;

*2006;
proc sort data = y2006_1; by id; run;
proc sort data = y2006_3; by id; run;
data y2006_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2006_1 y2006_3;
by id;
run;

*2007;
proc sort data = y2007_1; by id; run;
proc sort data = y2007_3; by id; run;
data y2007_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2007_1 y2007_3;
by id;
run;

*2008;
proc sort data = y2008_1; by id; run;
proc sort data = y2008_3; by id; run;
data y2008_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2008_1 y2008_3;
by id;
run;

*2009;
proc sort data = y2009_1; by id; run;
proc sort data = y2009_3; by id; run;
```

```
data y2009_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2009_1 y2009_3;
by id;
run;

*2010;
proc sort data = y2010_1; by id; run;
proc sort data = y2010_3; by id; run;
data y2010_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2010_1 y2010_3;
by id;
run;

*2011;
proc sort data = y2011_1; by id; run;
proc sort data = y2011_3; by id; run;
data y2011_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2011_1 y2011_3;
by id;
run;

*2012;
proc sort data = y2012_1; by id; run;
proc sort data = y2012_3; by id; run;
data y2012_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2012_1 y2012_3;
by id;
run;

*2013;
proc sort data = y2013_1; by id; run;
proc sort data = y2013_3; by id; run;
data y2013_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2013_1 y2013_3;
by id;
run;

*2014;
proc sort data = y2014_1; by id; run;
proc sort data = y2014_3; by id; run;
data y2014_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2014_1 y2014_3;
by id;
run;

*2015;
proc sort data = y2015_1; by id; run;
proc sort data = y2015_3; by id; run;
data y2015_12 (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
merge y2015_1 y2015_3;
by id;
run;


***************************************************************************************************************************
```

```
****************************************************************************************************************************
****************************************************************************************************************************
                                                                                         *8th/10th GRADE*
****************************************************************************************************************************
****************************************************************************************************************************
****************************************************************************************************************************
;
%macro eightTensetup(outfile, infile, year, grade, id, form, sex, mother_ed, father_ed, evenings,
taken, building, damaged, takenl, gang, fight, hurt);

data work.&outfile;
set eigthten.&infile;

year=&year;
grade=&grade;
id=&id;
form=&form;
sex=&sex; *1=male, 2=female;
if &sex = -9 or &sex = 9 then sex = .;

if &mother_ed in (1,2) then mother_ed = 1; *some highschool or lower;
if &mother_ed = 3 then mother_ed =2; *finished highschool;
if &mother_ed in (4,5,6) then mother_ed = 3; *some college to graduate school;
if &mother_ed = 7 then mother_ed = 0; *don't know;

if &father_ed in (1,2) then father_ed = 1; *some highschool or lower;
if &father_ed = 3 then father_ed =2; *finished highschool;
if &father_ed in (4,5,6) then father_ed = 3; *some college to graduate school;
if &father_ed = 7 then father_ed = 0; *don't know;
evenings=&evenings;*1=<1, 2=one, 3=two, 4=three, 5=four-five, 6=six-seven;

*CP;
taken=&taken;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if &taken = 9 or &taken = -9 then taken =.;
building=&building;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if &building = 9 or &building = -9 then building =.;
damaged=&damaged;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if &damaged = 9 or &damaged = -9 then damaged =.;
takenl=&takenl;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if &takenl = 9 or &takenl = -9 then takenl =.;

gang=&gang;     *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if &gang = 9 or &gang = -9 then gang =.;
fight=&fight; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if &fight = 9 or &fight = -9 then fight =.;
hurt=&hurt; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if &hurt = 9 or &hurt = -9 then hurt =.;

run;


%mend;

%eightTensetup(y1991_2, y1991_f2, 1991, 8, v4, 2, v2226, v2240, v2239, v2263, v2366, v2367, v2368, v2365, v2363, v2362, v2364);
%eightTensetup(y1991_4, y1991_f4, 1991, 10, v4, 2, v2226, v2240, v2239, v2263, v2366, v2367, v2368, v2365, v2363, v2362, v2364);
```

```
%eightTensetup(y1992_2, y1992_f2, 1992, 8, v4, 2, v2224, v2238, v2237, v2261, v2366, v2367, v2368, v2365, v2363, v2362, v2364);
%eightTensetup(y1992_4, y1992_f4, 1992, 10, v4, 2, v2224, v2238, v2237, v2261, v2366, v2367, v2368, v2365, v2363, v2362, v2364);
%eightTensetup(y1993_2, y1993_f2, 1993, 8, v4, 2, v2227, v2241, v2240, v2264, v2374, v2375, v2376, v2373, v2371, v2370, v2372);
%eightTensetup(y1993_4, y1993_f4, 1993, 10, v4, 2, v2227, v2241, v2240, v2264, v2374, v2375, v2376, v2373, v2371, v2370, v2372);
%eightTensetup(y1994_2, y1994_f2, 1994, 8, v4, 2, v2227, v2241, v2240, v2264, v2374, v2375, v2376, v2373, v2371, v2370, v2372);
%eightTensetup(y1994_4, y1994_f4, 1994, 10, v4, 2, v2227, v2241, v2240, v2264, v2374, v2375, v2376, v2373, v2371, v2370, v2372);
%eightTensetup(y1995_2, y1995_f2, 1995, 8, v4, 2, v2228, v2242, v2241, v2265, v2375, v2376, v2377, v2374, v2372, v2371, v2373);
%eightTensetup(y1995_4, y1995_f4, 1995, 10, v4, 2, v2228, v2242, v2241, v2265, v2375, v2376, v2377, v2374, v2372, v2371, v2373);
%eightTensetup(y1996_2, y1996_f2, 1996, 8, v4, 2, v2233, v2247, v2246, v2270, v2380, v2381, v2382, v2379, v2377, v2376, v2378);
%eightTensetup(y1996_4, y1996_f4, 1996, 10, v4, 2, v2233, v2247, v2246, v2270, v2380, v2381, v2382, v2379, v2377, v2376, v2378);
%eightTensetup(y1997_2, y1997_f2, 1997, 8, v4, 2, v2233, v2247, v2246, v2270, v2380, v2381, v2382, v2379, v2377, v2376, v2378);
%eightTensetup(y1997_6, y1997_f6, 1997, 10, v4, 2, v2233, v2247, v2246, v2270, v2380, v2381, v2382, v2379, v2377, v2376, v2378);
%eightTensetup(y1998_2, y1998_f2, 1998, 8, v4, 2, v2233, v2247, v2246, v2270, v2380, v2381, v2382, v2379, v2377, v2376, v2378);
%eightTensetup(y1998_6, y1998_f6, 1998, 10, v4, 2, v2233, v2247, v2246, v2270, v2380, v2381, v2382, v2379, v2377, v2376, v2378);
%eightTensetup(y1999_2, y1999_f2, 1999, 8, v4, 2, v2233, v2247, v2246, v2270, v2374, v2375, v2376, v2373, v2371, v2370, v2372);
%eightTensetup(y1999_6, y1999_f6, 1999, 10, v4, 2, v2233, v2247, v2246, v2270, v2374, v2375, v2376, v2373, v2371, v2370, v2372);
%eightTensetup(y2000_2, y2000_f2, 2000, 8, v4, 2, v2233, v2247, v2246, v2270, v2378, v2379, v2380, v2377, v2375, v2374, v2376);
%eightTensetup(y2000_6, y2000_f6, 2000, 10, v4, 2, v2233, v2247, v2246, v2270, v2378, v2379, v2380, v2377, v2375, v2374, v2376);
%eightTensetup(y2001_2, y2001_f2, 2001, 8, v4, 2, v2236, v2250, v2249, v2273, v2381, v2382, v2383, v2380, v2378, v2377, v2379);
%eightTensetup(y2001_6, y2001_f6, 2001, 10, v4, 2, v2236, v2250, v2249, v2273, v2381, v2382, v2383, v2380, v2378, v2377, v2379);
%eightTensetup(y2002_2, y2002_f2, 2002, 8, v4, 2, v2236, v2250, v2249, v2273, v2381, v2382, v2383, v2380, v2378, v2377, v2379);
%eightTensetup(y2002_6, y2002_f6, 2002, 10, v4, 2, v2236, v2250, v2249, v2273, v2381, v2382, v2383, v2380, v2378, v2377, v2379);
%eightTensetup(y2003_2, y2003_f2, 2003, 8, v4, 2, v2236, v2250, v2249, v2273, v2381, v2382, v2383, v2380, v2378, v2377, v2379);
%eightTensetup(y2003_6, y2003_f6, 2003, 10, v4, 2, v2236, v2250, v2249, v2273, v2381, v2382, v2383, v2380, v2378, v2377, v2379);
%eightTensetup(y2004_2, y2004_f2, 2004, 8, v4, 2, v2237, v2251, v2250, v2274, v2382, v2383, v2384, v2381, v2379, v2378, v2380);
%eightTensetup(y2004_6, y2004_f6, 2004, 10, v4, 2, v2237, v2251, v2250, v2274, v2382, v2383, v2384, v2381, v2379, v2378, v2380);
%eightTensetup(y2005_2, y2005_f2, 2005, 8, v4, 2, v2237, v2251, v2250, v2274, v2382, v2383, v2384, v2381, v2379, v2378, v2380);
%eightTensetup(y2005_6, y2005_f6, 2005, 10, v4, 2, v2237, v2251, v2250, v2274, v2382, v2383, v2384, v2381, v2379, v2378, v2380);
%eightTensetup(y2006_2, y2006_f2, 2006, 8, v4, 2, v2237, v2251, v2250, v2274, v2382, v2383, v2384, v2381, v2379, v2378, v2380);
%eightTensetup(y2006_6, y2006_f6, 2006, 10, v4, 2, v2237, v2251, v2250, v2274, v2382, v2383, v2384, v2381, v2379, v2378, v2380);
%eightTensetup(y2007_2, y2007_f2, 2007, 8, v4, 2, v2237, v2251, v2250, v2274, v2382, v2383, v2384, v2381, v2379, v2378, v2380);
%eightTensetup(y2007_6, y2007_f6, 2007, 10, v4, 2, v2237, v2251, v2250, v2274, v2382, v2383, v2384, v2381, v2379, v2378, v2380);
%eightTensetup(y2008_2, y2008_f2, 2008, 8, v4, 2, v2237, v2251, v2250, v2274, v2383, v2384, v2385, v2382, v2380, v2379, v2381);
%eightTensetup(y2008_6, y2008_f6, 2008, 10, v4, 2, v2237, v2251, v2250, v2274, v2383, v2384, v2385, v2382, v2380, v2379, v2381);
%eightTensetup(y2009_2, y2009_f2, 2009, 8, v4, 2, v2238, v2252, v2251, v2275, v2384, v2385, v2386, v2383, v2381, v2380, v2382);
%eightTensetup(y2009_6, y2009_f6, 2009, 10, v4, 2, v2238, v2252, v2251, v2275, v2384, v2385, v2386, v2383, v2381, v2380, v2382);
%eightTensetup(y2010_2, y2010_f2, 2010, 8, v4, 2, v2238, v2252, v2251, v2275, v2384, v2385, v2386, v2383, v2381, v2380, v2382);
%eightTensetup(y2010_6, y2010_f6, 2010, 10, v4, 2, v2238, v2252, v2251, v2275, v2384, v2385, v2386, v2383, v2381, v2380, v2382);
%eightTensetup(y2011_2, y2011_f2, 2011, 8, v4, 2, v2238, v2252, v2251, v2275, v2384, v2385, v2386, v2383, v2381, v2380, v2382);
%eightTensetup(y2011_6, y2011_f6, 2011, 10, v4, 2, v2238, v2252, v2251, v2275, v2384, v2385, v2386, v2383, v2381, v2380, v2382);

*** Datasets from 2012 - 2015 for 8/10th graders were created
differently so a simple data step was used instead of a macro  ***


*2012;
*8th;
data work.y2012_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set eigthten.y2012_8th10th;

year=2012;
grade = v501;
if v501 = 10 then delete;
id=v4;
form=v3;
```

```
sex= v7202; *1=male, 2=female;
if v7202 = -9 or v7202 = 9 then sex = .;

if v7216 in (1,2) then mother_ed = 1; *some highschool or lower;
if v7216 = 3 then mother_ed =2; *finished highschool;
if v7216 in (4,5,6) then mother_ed = 3; *some college to graduate school;
if v7216 = 7 then mother_ed = 0; *don't know;

if v7215 in (1,2) then father_ed = 1; *some highschool or lower;
if v7215 = 3 then father_ed =2; *finished highschool;
if v7215 in (4,5,6) then father_ed = 3; *some college to graduate school;
if v7215 = 7 then father_ed = 0; *don't know;
evenings=v7239;*1=<1, 2=one, 3=two, 4=three, 5=four-five, 6=six-seven;

*CP;
taken=v8520;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8520 = -8 or v8520 = -9 then taken =.;
building=v8521;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8521 = -8 or v8521 = -9 then building =.;
damaged=v8522;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8522 = -8 or v8522 = -9 then damaged =.;
takenl=v8519;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8519 = 9 or v8519 = -9 then takenl =.;

gang=v8517;     *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8517 = 9 or v8517 = -9 then gang =.;
fight=v8516;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8516 = 9 or v8516 = -9 then fight =.;
hurt=v8518; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8518 = -8 or v8518 = -9 then hurt =.;
run;

*10th;
data work.y2012_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set eigthten.y2012_8th10th;

year=2012;
grade = v501;
if v501 = 8 then delete;
id=v4;
form=v3;
sex= v7202; *1=male, 2=female;
if v7202 = -9 or v7202 = 9 then sex = .;

if v7216 in (1,2) then mother_ed = 1; *some highschool or lower;
if v7216 = 3 then mother_ed =2; *finished highschool;
if v7216 in (4,5,6) then mother_ed = 3; *some college to graduate school;
if v7216 = 7 then mother_ed = 0; *don't know;

if v7215 in (1,2) then father_ed = 1; *some highschool or lower;
if v7215 = 3 then father_ed =2; *finished highschool;
if v7215 in (4,5,6) then father_ed = 3; *some college to graduate school;
if v7215 = 7 then father_ed = 0; *don't know;
evenings=v7239;*1=<1, 2=one, 3=two, 4=three, 5=four-five, 6=six-seven;
```

```
*CP;
taken=v8520;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8520 = -8 or v8520 = -9 then taken =.;
building=v8521;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8521 = -8 or v8521 = -9 then building =.;
damaged=v8522;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8522 = -8 or v8522 = -9 then damaged =.;
takenl=v8519;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8519 = 9 or v8519 = -9 then takenl =.;

gang=v8517;     *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8517 = 9 or v8517 = -9 then gang =.;
fight=v8516; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8516 = 9 or v8516 = -9 then fight =.;
hurt=v8518; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8518 = -8 or v8518 = -9 then hurt =.;
run;

*2013;
*8th;
data work.y2013_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set eigthten.y2013_8th10th;

year=2013;
grade=v501;
if v501 = 10 then delete;
id=v4;
form=v3;
sex= v7202; *1=male, 2=female;
if v7202 = -9 or v7202 = 9 then sex = .;

if v7216 in (1,2) then mother_ed = 1; *some highschool or lower;
if v7216 = 3 then mother_ed =2; *finished highschool;
if v7216 in (4,5,6) then mother_ed = 3; *some college to graduate school;
if v7216 = 7 then mother_ed = 0; *don't know;

if v7215 in (1,2) then father_ed = 1; *some highschool or lower;
if v7215 = 3 then father_ed =2; *finished highschool;
if v7215 in (4,5,6) then father_ed = 3; *some college to graduate school;
if v7215 = 7 then father_ed = 0; *don't know;
evenings=v7239;*1=<1, 2=one, 3=two, 4=three, 5=four-five, 6=six-seven;

*CP;
taken=v8520;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8520 = -8 or v8520 = -9 then taken =.;
building=v8521;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8521 = -8 or v8521 = -9 then building =.;
damaged=v8522;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8522 = -8 or v8522 = -9 then damaged =.;
takenl=v8519;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8519 = 9 or v8519 = -9 then takenl =.;

gang=v8517;     *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8517 = 9 or v8517 = -9 then gang =.;
fight=v8516;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
```

```
if v8516 = 9 or v8516 = -9 then fight =.;
hurt=v8518; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8518 = -8 or v8518 = -9 then hurt =.;
run;

*10th;
data work.y2013_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set eigthten.y2013_8th10th;

year=2013;
grade=v501;
if v501 = 8 then delete;
id=v4;
form=v3;
sex= v7202; *1=male, 2=female;
if v7202 = -9 or v7202 = 9 then sex = .;

if v7216 in (1,2) then mother_ed = 1; *some highschool or lower;
if v7216 = 3 then mother_ed =2; *finished highschool;
if v7216 in (4,5,6) then mother_ed = 3; *some college to graduate school;
if v7216 = 7 then mother_ed = 0; *don't know;

if v7215 in (1,2) then father_ed = 1; *some highschool or lower;
if v7215 = 3 then father_ed =2; *finished highschool;
if v7215 in (4,5,6) then father_ed = 3; *some college to graduate school;
if v7215 = 7 then father_ed = 0; *don't know;
evenings=v7239;*1=<1, 2=one, 3=two, 4=three, 5=four-five, 6=six-seven;

*CP;
taken=v8520;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8520 = -8 or v8520 = -9 then taken =.;
building=v8521;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8521 = -8 or v8521 = -9 then building =.;
damaged=v8522;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8522 = -8 or v8522 = -9 then damaged =.;
takenl=v8519;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8519 = 9 or v8519 = -9 then takenl =.;

gang=v8517;     *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8517 = 9 or v8517 = -9 then gang =.;
fight=v8516; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8516 = 9 or v8516 = -9 then fight =.;
hurt=v8518; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8518 = -8 or v8518 = -9 then hurt =.;
run;

*2014;
*8th;
data work.y2014_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set eigthten.y2014_8th10th;

year=2014;
grade=v501;
if v501 = 10 then delete;
id=v4;
```

```
form=v3;
sex= v7202; *1=male, 2=female;
if v7202 = -9 or v7202 = 9 then sex = .;

if v7216 in (1,2) then mother_ed = 1; *some highschool or lower;
if v7216 = 3 then mother_ed =2; *finished highschool;
if v7216 in (4,5,6) then mother_ed = 3; *some college to graduate school;
if v7216 = 7 then mother_ed = 0; *don't know;

if v7215 in (1,2) then father_ed = 1; *some highschool or lower;
if v7215 = 3 then father_ed =2; *finished highschool;
if v7215 in (4,5,6) then father_ed = 3; *some college to graduate school;
if v7215 = 7 then father_ed = 0; *don't know;
evenings=v7239;*1=<1, 2=one, 3=two, 4=three, 5=four-five, 6=six-seven;

*CP;
taken=v8520;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8520 = -8 or v8520 = -9 then taken =.;
building=v8521;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8521 = -8 or v8521 = -9 then building =.;
damaged=v8522;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8522 = -8 or v8522 = -9 then damaged =.;
takenl=v8519;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8519 = 9 or v8519 = -9 then takenl =.;

gang=v8517;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8517 = 9 or v8517 = -9 then gang =.;
fight=v8516;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8516 = 9 or v8516 = -9 then fight =.;
hurt=v8518; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8518 = -8 or v8518 = -9 then hurt =.;
run;

*10th;
data work.y2014_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set eigthten.y2014_8th10th;

year=2014;
grade=v501;
if v501 = 8 then delete;
id=v4;
form=v3;
sex= v7202; *1=male, 2=female;
if v7202 = -9 or v7202 = 9 then sex = .;

if v7216 in (1,2) then mother_ed = 1; *some highschool or lower;
if v7216 = 3 then mother_ed =2; *finished highschool;
if v7216 in (4,5,6) then mother_ed = 3; *some college to graduate school;
if v7216 = 7 then mother_ed = 0; *don't know;

if v7215 in (1,2) then father_ed = 1; *some highschool or lower;
if v7215 = 3 then father_ed =2; *finished highschool;
if v7215 in (4,5,6) then father_ed = 3; *some college to graduate school;
if v7215 = 7 then father_ed = 0; *don't know;
evenings=v7239;*1=<1, 2=one, 3=two, 4=three, 5=four-five, 6=six-seven;
```

```
*CP;
taken=v8520;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8520 = -8 or v8520 = -9 then taken =.;
building=v8521;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8521 = -8 or v8521 = -9 then building =.;
damaged=v8522;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8522 = -8 or v8522 = -9 then damaged =.;
takenl=v8519;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8519 = 9 or v8519 = -9 then takenl =.;

gang=v8517;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8517 = 9 or v8517 = -9 then gang =.;
fight=v8516;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8516 = 9 or v8516 = -9 then fight =.;
hurt=v8518; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8518 = -8 or v8518 = -9 then hurt =.;
run;

*2015;
*8th;
data work.y2015_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set eigthten.y2015_8th10th;

year=2015;
grade=v501;
if v501 = 10 then delete;
id=v4;
form=v3;
sex= v7202; *1=male, 2=female;
if v7202 = -9 or v7202 = 9 then sex = .;

if v7216 in (1,2) then mother_ed = 1; *some highschool or lower;
if v7216 = 3 then mother_ed =2; *finished highschool;
if v7216 in (4,5,6) then mother_ed = 3; *some college to graduate school;
if v7216 = 7 then mother_ed = 0; *don't know;

if v7215 in (1,2) then father_ed = 1; *some highschool or lower;
if v7215 = 3 then father_ed =2; *finished highschool;
if v7215 in (4,5,6) then father_ed = 3; *some college to graduate school;
if v7215 = 7 then father_ed = 0; *don't know;
evenings=v7239;*1=<1, 2=one, 3=two, 4=three, 5=four-five, 6=six-seven;

*CP;
taken=v8520;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8520 = -8 or v8520 = -9 then taken =.;
building=v8521;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8521 = -8 or v8521 = -9 then building =.;
damaged=v8522;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8522 = -8 or v8522 = -9 then damaged =.;
takenl=v8519;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8519 = 9 or v8519 = -9 then takenl =.;

gang=v8517;     *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8517 = 9 or v8517 = -9 then gang =.;
```

```
fight=v8516; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8516 = 9 or v8516 = -9 then fight =.;
hurt=v8518; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8518 = -8 or v8518 = -9 then hurt =.;
run;

*10th;
data work.y2015_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set eigthten.y2015_8th10th;

year=2015;
grade=v501;
if v501 = 8 then delete;
id=v4;
form=v3;
sex= v7202; *1=male, 2=female;
if v7202 = -9 or v7202 = 9 then sex = .;

if v7216 in (1,2) then mother_ed = 1; *some highschool or lower;
if v7216 = 3 then mother_ed =2; *finished highschool;
if v7216 in (4,5,6) then mother_ed = 3; *some college to graduate school;
if v7216 = 7 then mother_ed = 0; *don't know;

if v7215 in (1,2) then father_ed = 1; *some highschool or lower;
if v7215 = 3 then father_ed =2; *finished highschool;
if v7215 in (4,5,6) then father_ed = 3; *some college to graduate school;
if v7215 = 7 then father_ed = 0; *don't know;
evenings=v7239;*1=<1, 2=one, 3=two, 4=three, 5=four-five, 6=six-seven;

*CP;
taken=v8520;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8520 = -8 or v8520 = -9 then taken =.;
building=v8521;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8521 = -8 or v8521 = -9 then building =.;
damaged=v8522;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8522 = -8 or v8522 = -9 then damaged =.;
takenl=v8519;*1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8519 = 9 or v8519 = -9 then takenl =.;

gang=v8517;     *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8517 = 9 or v8517 = -9 then gang =.;
fight=v8516; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8516 = 9 or v8516 = -9 then fight =.;
hurt=v8518; *1= Not at all, 2= Once, 3= Twice, 4= 3 or 4 Times, 5= 5 or more times;
if v8518 = -8 or v8518 = -9 then hurt =.;
run;

*Merging years for 8th/12th graders;
*1991;
proc sort data = y1991_2; by id; run;
proc sort data = y1991_4; by id; run;
data y1991_e(keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1991_2;
run;
data y1991_t(keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
```

```
set y1991_4;
run;

*1992;
proc sort data = y1992_2; by id; run;
proc sort data = y1992_4; by id; run;
data y1992_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1992_2;
run;
data y1992_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1992_4;
run;

*1993;
proc sort data = y1993_2; by id; run;
proc sort data = y1993_4; by id; run;
data y1993_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1993_2;
run;
data y1993_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1993_4;
run;

*1994;
proc sort data = y1994_2; by id; run;
proc sort data = y1994_4; by id; run;
data y1994_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1994_2;
run;
data y1994_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1994_4;
run;

*1995;
proc sort data = y1995_2; by id; run;
proc sort data = y1995_4; by id; run;
data y1995_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1995_2;
run;
data y1995_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1995_4;
run;

*1996;
proc sort data = y1996_2; by id; run;
proc sort data = y1996_4; by id; run;
data y1996_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1996_2;
run;
data y1996_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1996_4;
run;

*1997;
proc sort data = y1997_2; by id; run;
proc sort data = y1997_6; by id; run;
data y1997_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1997_2;
run;
data y1997_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1997_6;
run;

*1998;
proc sort data = y1998_2; by id; run;
proc sort data = y1998_6; by id; run;
data y1998_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1998_2;
run;
data y1998_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1998_6 ;
run;

*1999;
proc sort data = y1999_2; by id; run;
proc sort data = y1999_6; by id; run;
data y1999_e(keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1999_2;
run;
data y1999_t(keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y1999_6;
run;

*2000;
proc sort data = y2000_2; by id; run;
proc sort data = y2000_6; by id; run;
data y2000_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2000_2;
run;
data y2000_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2000_6;
run;

*2001;
proc sort data = y2001_2; by id; run;
proc sort data = y2001_6; by id; run;
data y2001_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2001_2;
run;
data y2001_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2001_6;
run;

*2002;
proc sort data = y2002_2; by id; run;
proc sort data = y2002_6; by id; run;
data y2002_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2002_2;
run;
data y2002_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
```

```
set y2002_6;
run;

*2003;
proc sort data = y2003_2; by id; run;
proc sort data = y2003_6; by id; run;
data y2003_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2003_2;
run;
data y2003_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2003_6;
run;

*2004;
proc sort data = y2004_2; by id; run;
proc sort data = y2004_6; by id; run;
data y2004_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2004_2;
run;
data y2004_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2004_6;
run;

*2005;
proc sort data = y2005_2; by id; run;
proc sort data = y2005_6; by id; run;
data y2005_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2005_2;
run;
data y2005_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2005_6;
run;

*2006;
proc sort data = y2006_2; by id; run;
proc sort data = y2006_6; by id; run;
data y2006_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2006_2;
run;
data y2006_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2006_6;
run;

*2007;
proc sort data = y2007_2; by id; run;
proc sort data = y2007_6; by id; run;
data y2007_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2007_2;
run;
data y2007_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2007_6;
run;

*2008;
proc sort data = y2008_2; by id; run;
proc sort data = y2008_6; by id; run;
data y2008_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2008_2;
run;
data y2008_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2008_6;
run;

*2009;
proc sort data = y2009_2; by id; run;
proc sort data = y2009_6; by id; run;
data y2009_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2009_2;
run;
data y2009_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2009_6;
run;

*2010;
proc sort data = y2010_2; by id; run;
proc sort data = y2010_6; by id; run;
data y2010_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2010_2;
run;
data y2010_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2010_6;
run;

*2011;
proc sort data = y2011_2; by id; run;
proc sort data = y2011_6; by id; run;
data y2011_e (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2011_2;
run;
data y2011_t (keep= year grade id form sex mother_ed father_ed taken building damaged takenl hurt gang fight evenings);
set y2011_6;
run;





*Merging age and 5 level race variable;
***************************************************************************************************************************
***************************************************************************************************************************
***************************************************************************************************************************
                                                                                        *MERGING AGE AND RACE*
***************************************************************************************************************************
***************************************************************************************************************************
***************************************************************************************************************************

************** 8th grade;
proc sort data = tim.k8th_9192; by year; run;
%macro timyears8(dataname, year);
data &dataname;
```

```
set tim.k8th_9192;
if year ~= &year then delete;
tim =1;
id=arch_id;
run;
proc sort data= &dataname; by id ;run;
%mend;

%timyears8(tim91_8, 1991);
%timyears8(tim92_8, 1992);

proc sort data = tim.k8th_final_3_11; by year; run;

*macro to break up tim data by year;
%macro timyears8(dataname, year);
data &dataname (drop = arch_id);
set tim.k8th_final_3_11;
if year ~= &year then delete;
id =arch_id;
tim = 1;
run;
proc sort data= &dataname; by id ;run;
%mend;

%timyears8(tim93_8, 1993);
%timyears8(tim94_8, 1994);
%timyears8(tim95_8, 1995);
%timyears8(tim97_8, 1997);
%timyears8(tim98_8, 1998);
%timyears8(tim99_8, 1999);
%timyears8(tim00_8, 2000);
%timyears8(tim01_8, 2001);
%timyears8(tim02_8, 2002);
%timyears8(tim03_8, 2003);
%timyears8(tim04_8, 2004);
%timyears8(tim05_8, 2005);
%timyears8(tim06_8, 2006);
%timyears8(tim07_8, 2007);
%timyears8(tim08_8, 2008);
%timyears8(tim09_8, 2009);
%timyears8(tim10_8, 2010);
%timyears8(tim11_8, 2011);
%timyears8(tim12_8, 2012);

data tim96_8 (drop = arch_id);
set tim.k8th_1996;
tim =1;
id = arch_id;
if year ~= 1996 then delete;
run;

*Adding in 2013/14);
proc sort data = tim.k8th_2014; by year; run;
%macro timyears8(dataname, year);
data &dataname;
```

```
set tim.k8th_2014;
if year ~= &year then delete;
tim =1;
id=arch_id;
run;
proc sort data= &dataname; by id ;run;
%mend;

%timyears8(tim13_8, 2013);
%timyears8(tim14_8, 2014);

*adding 2015;
proc sort data = tim.k8th_2015; by year; run;
%macro timyears8(dataname, year);
data &dataname;
set tim.k8th_2015;
if year ~= &year then delete;
tim =1;
id=arch_id;
run;
proc sort data= &dataname; by id ;run;
%mend;

%timyears8(tim15_8, 2015);

*macro to combine by year;
%macro ageMerge(dataset1, dataset2, finalYearSet);
proc sort data= &dataset1; by id ;run;
proc sort data= &dataset2; by id ;run;

data &finalYearSet;
merge &dataset1 &dataset2; by id; run;

run;

%mend;

%ageMerge(y1991_e, tim91_8, age8_91);
%ageMerge(y1992_e, tim92_8, age8_92);
%ageMerge(y1993_e, tim93_8, age8_93);
%ageMerge(y1994_e, tim94_8, age8_94);
%ageMerge(y1995_e, tim95_8, age8_95);
%ageMerge(y1996_e, tim96_8, age8_96);
%ageMerge(y1997_e, tim97_8, age8_97);
%ageMerge(y1998_e, tim98_8, age8_98);
%ageMerge(y1999_e, tim99_8, age8_99);
%ageMerge(y2000_e, tim00_8, age8_00);
%ageMerge(y2001_e, tim01_8, age8_01);
%ageMerge(y2002_e, tim02_8, age8_02);
%ageMerge(y2003_e, tim03_8, age8_03);
%ageMerge(y2004_e, tim04_8, age8_04);
%ageMerge(y2005_e, tim05_8, age8_05);
%ageMerge(y2006_e, tim06_8, age8_06);
%ageMerge(y2007_e, tim07_8, age8_07);
%ageMerge(y2008_e, tim08_8, age8_08);
```

```
%ageMerge(y2009_e, tim09_8, age8_09);
%ageMerge(y2010_e, tim10_8, age8_10);
%ageMerge(y2011_e, tim11_8, age8_11);
%ageMerge(y2012_e, tim12_8, age8_12);
%ageMerge(y2013_e, tim13_8, age8_13);
%ageMerge(y2014_e, tim14_8, age8_14);
%ageMerge(y2015_e, tim15_8, age8_15);

************** 10th grade;
proc sort data = tim.k10th_9192; by year; run;
%macro timyears10(dataname, year);
data &dataname;
set tim.k10th_9192;
if year ~= &year then delete;
tim =1;
id=arch_id;
run;
proc sort data= &dataname; by id grade;run;
%mend;

%timyears10(tim91_10, 1991);
%timyears10(tim92_10, 1992);

proc sort data = tim.k10thbmi; by year; run;

%macro timyears10(dataname, year);
data &dataname (drop = arch_id);
set tim.k10thbmi ;
if year ~= &year then delete;
id =arch_id;
tim = 1;
run;
proc sort data= &dataname; by id grade;run;
%mend;

%timyears10(tim93_10, 1993);
%timyears10(tim94_10, 1994);
%timyears10(tim95_10, 1995);
%timyears10(tim97_10, 1997);
%timyears10(tim98_10, 1998);
%timyears10(tim99_10, 1999);
%timyears10(tim00_10, 2000);
%timyears10(tim01_10, 2001);
%timyears10(tim02_10, 2002);
%timyears10(tim03_10, 2003);
%timyears10(tim04_10, 2004);
%timyears10(tim05_10, 2005);
%timyears10(tim06_10, 2006);
%timyears10(tim07_10, 2007);
%timyears10(tim08_10, 2008);
%timyears10(tim09_10, 2009);
%timyears10(tim10_10, 2010);
%timyears10(tim11_10, 2011);
%timyears10(tim12_10, 2012);
```

```
data tim96_10;
set tim.k10th_1996_0407;
tim =1;
id = arch_id;
if year ~= 1996 then delete;
run;

*Adding in 2013/14);
proc sort data = tim.k10th_2014; by year; run;
%macro timyears8(dataname, year);
data &dataname;
set tim.k10th_2014;
if year ~= &year then delete;
tim =1;
id=arch_id;
run;
proc sort data= &dataname; by id ;run;
%mend;

%timyears8(tim13_10, 2013);
%timyears8(tim14_10, 2014);

*adding 2015;
proc sort data = tim.k10th_2015; by year; run;
%macro timyears8(dataname, year);
data &dataname;
set tim.k10th_2015;
if year ~= &year then delete;
tim =1;
id=arch_id;
run;
proc sort data= &dataname; by id ;run;
%mend;

%timyears8(tim15_10, 2015);

*macro to break up tim data by year;
%macro ageMerge(dataset1, dataset2, finalYearSet);
proc sort data= &dataset1; by id grade;run;
proc sort data= &dataset2; by id grade;run;

data &finalYearSet;
merge &dataset1 &dataset2; by id grade; run;

run;

%mend;

%ageMerge(y1991_t, tim91_10, race10_91);
%ageMerge(y1992_t, tim92_10, race10_92);
%ageMerge(y1993_t, tim93_10, race10_93);
%ageMerge(y1994_t, tim94_10, race10_94);
%ageMerge(y1995_t, tim95_10, race10_95);
%ageMerge(y1996_t, tim96_10, race10_96);
%ageMerge(y1997_t, tim97_10, race10_97);
```

```
%ageMerge(y1998_t, tim98_10, race10_98);
%ageMerge(y1999_t, tim99_10, race10_99);
%ageMerge(y2000_t, tim00_10, race10_00);
%ageMerge(y2001_t, tim01_10, race10_01);
%ageMerge(y2002_t, tim02_10, race10_02);
%ageMerge(y2003_t, tim03_10, race10_03);
%ageMerge(y2004_t, tim04_10, race10_04);
%ageMerge(y2005_t, tim05_10, race10_05);
%ageMerge(y2006_t, tim06_10, race10_06);
%ageMerge(y2007_t, tim07_10, race10_07);
%ageMerge(y2008_t, tim08_10, race10_08);
%ageMerge(y2009_t, tim09_10, race10_09);
%ageMerge(y2010_t, tim10_10, race10_10);
%ageMerge(y2011_t, tim11_10, race10_11);
%ageMerge(y2012_t, tim12_10, race10_12);
%ageMerge(y2013_t, tim13_10, race10_13);
%ageMerge(y2014_t, tim14_10, race10_14);
%ageMerge(y2015_t, tim15_10, race10_15);

************** 12th grade;
proc sort data = tim.k12th_Final_3_11; by year; run;

*macro to break up tim data by year;
%macro timyears12(dataname, year);
data &dataname (drop = arch_id);
set tim.k12th_Final_3_11;
if year ~= &year then delete;
id = arch_id;
grade = 12;
tim= 1;
run;
proc sort data= &dataname; by id grade;run;
%mend;

%timyears12(tim91_12, 1991);
%timyears12(tim92_12, 1992);
%timyears12(tim93_12, 1993);
%timyears12(tim94_12, 1994);
%timyears12(tim95_12, 1995);
%timyears12(tim96_12, 1996);
%timyears12(tim97_12, 1997);
%timyears12(tim98_12, 1998);
%timyears12(tim99_12, 1999);
%timyears12(tim00_12, 2000);
%timyears12(tim01_12, 2001);
%timyears12(tim02_12, 2002);
%timyears12(tim03_12, 2003);
%timyears12(tim04_12, 2004);
%timyears12(tim05_12, 2005);
%timyears12(tim06_12, 2006);
%timyears12(tim07_12, 2007);
%timyears12(tim08_12, 2008);
%timyears12(tim09_12, 2009);
%timyears12(tim10_12, 2010);
%timyears12(tim11_12, 2011);
```

```
%timyears12(tim12_12, 2012);

data tim12_12;
set tim.k12th;
tim =1;
id = arch_id;
if year ~= 2012 then delete;
run;

*Adding in 2013/14;
proc sort data = tim.k12th_2014; by year; run;
%macro timyears8(dataname, year);
data &dataname;
set tim.k12th_2014;
if year ~= &year then delete;
tim =1;
id=arch_id;
run;
proc sort data= &dataname; by id ;run;
%mend;

%timyears8(tim13_12, 2013);
%timyears8(tim14_12, 2014);

*adding 2015;
proc sort data = tim.k12th_2015; by year; run;
%macro timyears8(dataname, year);
data &dataname;
set tim.k12th_2015;
if year ~= &year then delete;
tim =1;
id=arch_id;
run;
proc sort data= &dataname; by id ;run;
%mend;

%timyears8(tim15_12, 2015);

*macro to break up tim data by year;
%macro ageMerge(dataset1, dataset2, finalYearSet);
proc sort data= &dataset1; by id ;run;
proc sort data= &dataset2; by id ;run;

data &finalYearSet;
merge &dataset1 &dataset2; by id ; run;

run;

%mend;

%ageMerge(y1991_12, tim91_12, race12_91);
%ageMerge(y1992_12, tim92_12, race12_92);
%ageMerge(y1993_12, tim93_12, race12_93);
%ageMerge(y1994_12, tim94_12, race12_94);
%ageMerge(y1995_12, tim95_12, race12_95);
```

```
%ageMerge(y1996_12, tim96_12, race12_96);
%ageMerge(y1997_12, tim97_12, race12_97);
%ageMerge(y1998_12, tim98_12, race12_98);
%ageMerge(y1999_12, tim99_12, race12_99);
%ageMerge(y2000_12, tim00_12, race12_00);
%ageMerge(y2001_12, tim01_12, race12_01);
%ageMerge(y2002_12, tim02_12, race12_02);
%ageMerge(y2003_12, tim03_12, race12_03);
%ageMerge(y2004_12, tim04_12, race12_04);
%ageMerge(y2005_12, tim05_12, race12_05);
%ageMerge(y2006_12, tim06_12, race12_06);
%ageMerge(y2007_12, tim07_12, race12_07);
%ageMerge(y2008_12, tim08_12, race12_08);
%ageMerge(y2009_12, tim09_12, race12_09);
%ageMerge(y2010_12, tim10_12, race12_10);
%ageMerge(y2011_12, tim11_12, race12_11);
%ageMerge(y2012_12, tim12_12, race12_12);
%ageMerge(y2013_12, tim13_12, race12_13);
%ageMerge(y2014_12, tim14_12, race12_14);
%ageMerge(y2015_12, tim15_12, race12_15);

*Merging each grade together;
data CP_8thtwo;
set age8_91 age8_92 age8_93 age8_94 age8_95 age8_96 age8_97
age8_98 age8_99 age8_00 age8_01 age8_02 age8_03 age8_04 age8_05
age8_06 age8_07 age8_08 age8_09 age8_10 age8_11 age8_12 age8_13 age8_14 age8_15;
run;

data CP_10thtwo;
set race10_91 race10_92 race10_93 race10_94 race10_95 race10_96 race10_97
race10_98 race10_99  race10_00 race10_01 race10_02 race10_03 race10_04
race10_05 race10_06 race10_07 race10_08 race10_09 race10_10 race10_11 race10_12 race10_13 race10_14 race10_15;
run;

data CP_12thtwo;
set race12_91 race12_92 race12_93 race12_94 race12_95 race12_96 race12_97
race12_98 race12_99  race12_00 race12_01 race12_02 race12_03 race12_04
race12_05 race12_06 race12_07 race12_08 race12_09 race12_10 race12_11 race12_12 race12_13 race12_14 race12_15;
run;

*Final Dataset in SAS dropping 13 and 19 year olds and as well as unnecessary variables and marking missing values;
data tim.caroline_1_30;
set CP_12thtwo CP_10thtwo CP_8thtwo;
drop ARCH_ID BMI BMIPCT;
if taken = -9 or taken = 9 then taken =.;
if building = -9 or building = 9 then building =.;
if damaged = -9 or damaged = 9 then damaged = .;
if takenl = -9 or takenl = 9 then takenl = .;
if gang = -9 or gang = 9 then gang =.;
if fight = -9 or fight = 9 then fight = .;
if hurt = -9 or hurt = 9 then hurt =.;
if evenings = -9 or evenings = 9 then evenings = .;
if form ~ in (2,6) then delete;
if age < 13 then delete;
if age >=19 then delete;
```

```
run;

**********************STAT TRANSFER TO STATA FILE**************************************

use "/Users/cr2809/Dropbox/Conduct RR/caroline_1_30.dta"

* setting style of data
mi set wide

*registering imputed and regular variables
mi register imputed sex mother_ed father_ed evenings taken building damaged takenl fight gang hurt Race
mi register regular year grade Age

*impution step
mi impute chained (ologit, augment) mother_ed father_ed evenings taken building damaged takenl fight gang hurt (mlogit, augment) sex Race, add(10) rseed(1234)

ssc install apc

*imputed data saved here
use "/Users/cr2809/Dropbox/MTF2015/caroline_1_30_imputed.dta"

*generating transformed variables
mi passive: gen ager = floor(Age)
mi passive: egen yearcat = cut(year), at(1991,1996,2001,2006,2011,2016) icodes
mi passive: gen birthcoh = (year-ager)
mi passive: egen parent_ed = rowmax(father_ed mother_ed)
mi passive: egen parent_ed_cat = cut(parent_ed), at(1,3,4) icodes
mi passive: gen TPDscore = taken + building + damaged + takenl
mi passive: gen IAscore = fight + gang + hurt
mi passive: gen Overallscore = TPDscore + IAscore
*binaries of conduct items
mi passive: egen hurt_cat = cut(hurt), at(1,2,6) icodes
mi passive: egen gang_cat = cut(gang), at(1,2,6) icodes
mi passive: egen fight_cat = cut(fight), at(1,2,6) icodes
mi passive: egen taken_cat = cut(taken), at(1,2,6) icodes
mi passive: egen takenl_cat = cut(takenl), at(1,2,6) icodes
mi passive: egen building_cat = cut(building), at(1,2,6) icodes
mi passive: egen damaged_cat = cut(damaged), at((1,2,6) icodes

*AGE PERIOD COHORT MODELS
*Overall score by sex
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==1, age (ager) period (year) cohort (birthcoh)
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==2, age (ager) period (year) cohort (birthcoh)
*IA score by sex
mi estimate, esampvaryok cmdok: apc_ie IAscore if sex==1, age (ager) period (year) cohort (birthcoh)
mi estimate, esampvaryok cmdok: apc_ie IAscore if sex==2, age (ager) period (year) cohort (birthcoh)
*TPD score by sex
mi estimate, esampvaryok cmdok: apc_ie TPDscore if sex==1, age (ager) period (year) cohort (birthcoh)
mi estimate, esampvaryok cmdok: apc_ie TPDscore if sex==2, age (ager) period (year) cohort (birthcoh)
*Evenings out by sex
mi estimate, esampvaryok cmdok: apc_ie evenings if sex==1, age (ager) period (year) cohort (birthcoh)
mi estimate, esampvaryok cmdok: apc_ie evenings if sex==2, age (ager) period (year) cohort (birthcoh)
*Overall score by sex, white only
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==1 & Race==1, age (ager) period (year) cohort (birthcoh)
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==2 & Race==1, age (ager) period (year) cohort (birthcoh)
*Overall score by sex, black only
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==1 & Race==2, age (ager) period (year) cohort (birthcoh)
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==2 & Race==2, age (ager) period (year) cohort (birthcoh)
*Overall score by sex, hispanic only
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==1 & Race==3, age (ager) period (year) cohort (birthcoh)
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==2 & Race==3, age (ager) period (year) cohort (birthcoh)
*Overall score by sex, low parent education only
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==1 & parent_ed_cat==0, age (ager) period (year) cohort (birthcoh)
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==2 & parent_ed_cat==0, age (ager) period (year) cohort (birthcoh)
*Overall score by sex, high parent education only
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==1 & parent_ed_cat==1, age (ager) period (year) cohort (birthcoh)
mi estimate, esampvaryok cmdok: apc_ie Overallscore if sex==2 & parent_ed_cat==1, age (ager) period (year) cohort (birthcoh)

*MEANS
*by sex and year category
mi estimate: mean Overallscore, over(sex yearcat)
mi estimate: mean IAscore, over(sex yearcat)
mi estimate: mean TPDscore, over(sex yearcat)
*by sex, age, and year category
mi estimate: mean Overallscore, over(sex ager yearcat)
mi estimate: mean IAscore, over(sex ager yearcat)
mi estimate: mean TPDscore, over(sex ager yearcat)
*by sex
mi estimate: mean Overallscore, over(sex)
mi estimate: mean IAscore, over(sex)
mi estimate: mean TPDscore, over(sex)
*by sex and single year
mi estimate: mean Overallscore, over(sex year)
*evenings by sex, year, and grade
mi estimate: mean evenings, over(sex year grade)

*PROPORTIONS
*over sex, age, and year category
mi estimate: proportion building_cat, over(sex ager yearcat)
mi estimate: proportion damaged_cat, over(sex ager yearcat)
mi estimate: proportion fight_cat, over(sex ager yearcat)
mi estimate: proportion gang_cat, over(sex ager yearcat)
mi estimate: proportion taken_cat, over(sex ager yearcat)
mi estimate: proportion takenl_cat, over(sex ager yearcat)
mi estimate: proportion hurt_cat, over(sex ager yearcat)

*LINEAR REGRESSIONS
mi estimate: regress Overallscore evenings
mi estimate: regress IAscore evenings
mi estimate: regress TPDscore evenings
```