Supplemental Material - *Estimating Long-Term Pollution Exposure Effects Through Inverse Probability Weighting Methods with Cox Proportional Hazards Models*

S1. Background on Propensity Score Methods and Key Assumptions

Although not developed directly from propensity score methods, inverse probability weighting has a natural relation with propensity score methods.[1] Propensity score analysis was developed to control for non-randomization in treatment assignment in observational studies, where confounding variables may bias the estimated treatment effects by influencing both the exposure and the outcome of individuals.[2] Though often used in estimating the effect of binary exposure status to mimic the level of randomization found in a randomized control trial, propensity score methodologies have been expanded to include multi-valued and continuous treatments with the purpose of estimating average causal effects where treatment is uncorrelated with observed covariates.[3-5] Here, the treatment of interest is a continuous measure of average annual pollution levels, necessitating the use of propensity score methods that allow for greater flexibility in treatment assignment.

Several key assumptions are made in causal inference literature and are therefore key to the use of propensity score analysis: conditional exchangeability, positivity, consistency, and no interference.[5,6] The conditional exchangeability assumption posits that the adjustment for baseline covariate differences removes potential selection effects, and therefore eliminates bias in comparison by exposure level. Though this assumption applies to all possible confounders, it may only be verified for those which are measured; as such, multiple health and demographic variables are included to make this assumption plausible.[7] The second assumption, positivity, is that each individual in the study has a non-zero probability of receiving a particular exposure level of pollution given their measured covariates.[8] The plausibility of this assumption is

supported by the fact that every quartile of observed exposure contains individuals of all kinds (representing every categorical variable value and all numeric values such as age).

Consistency is also key to this analysis; this assumption may be phrased simply as that each individual's counterfactual outcome, under their observed exposure levels, is equal to their observed outcome.[9-10] Practically speaking, this means that it must be plausible that a person could have a different level of pollution exposure.[9] While certain factors like income and level of education may influence individual $PM_{2.5}$ exposure as both pollution and those demographics are correlated geographically, consistency is supported by the positivity assumption above and the inclusion of other covariates in the models estimated. Furthermore, exposure could conceivably though infeasibly be manipulated by individuals moving to areas with differing $PM_{2.5}$ concentrations without changing other individual-level factors. No interference, or the individualistic treatment response assumption, implies that the exposure status of a given individual does not have any bearing on the outcome of any other individual.[11] The cohort used in this study is comprised of individuals in the NHIS sample adult file, meaning that one adult is sampled from each family, and thus family-level correlations of treatment with outcomes (such as through the use of an air filter) will be absent in this cohort. Individuals sampled from the same census tract would have the same exposure value, but individual mortality is unlikely to be affected by the toxicity of the air that another breaths – only that which they personally breathe; while these exposure values are correlated, no interfering effect should exist.

S2. IPW Construction

The stabilized IPWs take the following form:

$$w = \frac{f_x(x;\theta)}{f_{x\,|\,C}(x \mid C = c;\theta)}$$

Here, x denotes the level of exposure, C denotes the measured potential confounders, and θ denotes the parameters of the relevant probability density functions, f_x and $f_{x \mid C}$.

S3. Model Specification

The models used in this analysis are given in the following equations:

$$\lambda(t \mid x) = \lambda_0(t) \exp(\beta_1 x) \quad (1)$$

$$\lambda(t \mid x, C) = \lambda_0(t) \exp(\beta_2 x + \gamma C') \quad (2)$$

The function λ represents the hazard function at time t, where λ_0 represents the baseline hazard for each age-sex-race category of the population. The estimated parameter β_1 represents the coefficient for a 10 μg/m^3 increase in annual average $PM_{2.5}$ exposure, x, while β_2 represents the estimated effect after adjustment for covariates. The terms C and γ are the row vectors for covariates and their estimated coefficients, respectively. While Cox models without IPWs typically follow the form of (2), both β_1 and β_2 in the weighted models are interpreted as causal effects rather than associations, though the assumption of no unmeasured confounders is inherently unverifiable.

SUPPLMENTAL REFERENCES

1 – Thoemmes F, Ong AD. A primer on inverse probability of treatment weighting and marginal structural models. *Emerg Adulthood*. 2015;4:40-59.

2 – Rosenbaum PR, Rubin DB. The central role of the propensity score in observational studies for causal effects. *Biometrika*. 1983;70:41-55.

3 – Imbens GW. The role of the propensity score in estimating dose-response functions. *Biometrika*. 2000;87:706-710.

4 – Hirano K, Imbens GW. The propensity score with continuous treatments. In: Gelman A, Meng XL eds. *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*. Wiley; 2004:73-84.

5 – Funk MJ, Westreich D, Wiesen C, Stürmer T, Brookhart MA, Davidian M. Doubly robust estimation of causal effects. *Am J Epidemiol*. 2011;173:761-767.

6 – Hernán MA, VanderWeele TJ. Compound treatments and transportability of causal inference. *Epidemiol*. 2011;22:368-377.

7 – Robins, JM. Association, causation, and marginal structural models. *Synthese*. 1999;121:151-179.

8 – Westreich D, Cole SR. Positivity in practice [Commentary]. *Am J Epidemiol*. 2010;171:674-677.

9 – Cole SR, Frangakis CE. The consistency statement in causal inference [Commentary]. *Epidemiol*. 2009;20:3-5.

10 – Cole SR, Hernán M. Constructing inverse probability weights for marginal structural models. *Am J Epidemiol*. 2008;168:656-664.

11 – Liu L, Hudgens MG, Becker-Dreps S. On inverse probability-weighted estimators in the presence of interference. *Biometrika*. 2016;103:829-842.

```
*************************************************************************************************;
*************************************************************************************************;
* SAS Code for IPW PM2.5-mortality analysis;
*************************************************************************************************;
*************************************************************************************************;


*************************************************************************************************;
* Data Preparation;
*************************************************************************************************;

libname z "Z:\Data";
libname temp "Z:\Data\Temp";

data ALL;
 set z.pope_final2;
run;

data FINAL;
 set ALL;

 *Assign wind speed values;
 wind_99to15 =.;
 wind_99to15 = wind_10_99to15;
 if wind_99to15 =. then wind_99to15 = wind_00_99to15;

 /*ASSIGN PM2.5 VALUES AND CALCULATE RATIO AND BACKCASTED VALUES*/
 pm25_1999 =.;   pm25_1999 = pm25_10_1999;   if pm25_1999 =. then pm25_1999 = pm25_00_1999;
 pm25_2000 =.;   pm25_2000 = pm25_10_2000;   if pm25_2000 =. then pm25_2000 = pm25_00_2000;
 pm25_2001 =.;   pm25_2001 = pm25_10_2001;   if pm25_2001 =. then pm25_2001 = pm25_00_2001;
 pm25_2002 =.;   pm25_2002 = pm25_10_2002;   if pm25_2002 =. then pm25_2002 = pm25_00_2002;
 pm25_2003 =.;   pm25_2003 = pm25_10_2003;   if pm25_2003 =. then pm25_2003 = pm25_00_2003;
 pm25_2004 =.;   pm25_2004 = pm25_10_2004;   if pm25_2004 =. then pm25_2004 = pm25_00_2004;
 pm25_2005 =.;   pm25_2005 = pm25_10_2005;   if pm25_2005 =. then pm25_2005 = pm25_00_2005;
 pm25_2006 =.;   pm25_2006 = pm25_10_2006;   if pm25_2006 =. then pm25_2006 = pm25_00_2006;
 pm25_2007 =.;   pm25_2007 = pm25_10_2007;   if pm25_2007 =. then pm25_2007 = pm25_00_2007;
 pm25_2008 =.;   pm25_2008 = pm25_10_2008;   if pm25_2008 =. then pm25_2008 = pm25_00_2008;
 pm25_2009 =.;   pm25_2009 = pm25_10_2009;   if pm25_2009 =. then pm25_2009 = pm25_00_2009;
 pm25_2010 =.;   pm25_2010 = pm25_10_2010;   if pm25_2010 =. then pm25_2010 = pm25_00_2010;
 pm25_2011 =.;   pm25_2011 = pm25_10_2011;   if pm25_2011 =. then pm25_2011 = pm25_00_2011;
 pm25_2012 =.;   pm25_2012 = pm25_10_2012;   if pm25_2012 =. then pm25_2012 = pm25_00_2012;
 pm25_2013 =.;   pm25_2013 = pm25_10_2013;   if pm25_2013 =. then pm25_2013 = pm25_00_2013;
 pm25_2014 =.;   pm25_2014 = pm25_10_2014;   if pm25_2014 =. then pm25_2014 = pm25_00_2014;
 pm25_2015 =.;   pm25_2015 = pm25_10_2015;   if pm25_2015 =. then pm25_2015 = pm25_00_2015;

 *Make classic "pm" variable, which will be equal to pm99to15.
  We make this variable since "pm" has been the standard in our regressions;
 pm = pm25_99to15/10;
run;

data FINAL;
 set FINAL;

 *Delete people from 2015 and those with unknown mortality status;
 if survey_year=2015 or MORTSTAT=. then delete;

 /*CREATE VARIABLE FOR URBAN RURAL*/
 *UR2KX is from 2010 Census Geocode File, and UR is from 2000 Census Geocoe File;
 length urban_rural $8.;
 if urb_rrl = 1 then urban_rural="2. Urban";
 if urb_rrl = 2 then urban_rural="1. Rural";

 /*CREATE VARIABLE FOR REGION*/
```

```
length region $12.;
if STATE2K="09" or STATE2K="23" or STATE2K="25" or STATE2K="33" or STATE2K="44" or
   STATE2K="50" or STATE2K="34" or STATE2K="36" or STATE2K="42"
 then region="1. Northeast";
if STATE2K="17" or STATE2K="18" or STATE2K="26" or STATE2K="39" or
   STATE2K="55" or STATE2K="19" or STATE2K="20" or STATE2K="27" or
   STATE2K="29" or STATE2K="31" or STATE2K="38" or STATE2K="46"
 then region="2. Midwest";
if STATE2K="10" or STATE2K="11" or STATE2K="12" or STATE2K="13" or STATE2K="24" or
   STATE2K="37" or STATE2K="45" or STATE2K="51" or STATE2K="54" or
   STATE2K="01" or STATE2K="21" or STATE2K="28" or STATE2K="47" or
   STATE2K="05" or STATE2K="22" or STATE2K="40" or STATE2K="48"
 then region="3. South";
if STATE2K="04" or STATE2K="08" or STATE2K="16" or STATE2K="30" or STATE2K="32" or
   STATE2K="35" or STATE2K="49" or STATE2K="56" or STATE2K="02" or
   STATE2K="06" or STATE2K="15" or STATE2K="41" or STATE2K="53"
 then region="4. West";
/*Repeat with 2KX*/
if STATE2KX="09" or STATE2KX="23" or STATE2KX="25" or STATE2KX="33" or STATE2KX="44" or
   STATE2KX="50" or STATE2KX="34" or STATE2KX="36" or STATE2KX="42"
 then region="1. Northeast";
if STATE2KX="17" or STATE2KX="18" or STATE2KX="26" or STATE2KX="39" or
   STATE2KX="55" or STATE2KX="19" or STATE2KX="20" or STATE2KX="27" or
   STATE2KX="29" or STATE2KX="31" or STATE2KX="38" or STATE2KX="46"
 then region="2. Midwest";
if STATE2KX="10" or STATE2KX="11" or STATE2KX="12" or STATE2KX="13" or STATE2KX="24" or
   STATE2KX="37" or STATE2KX="45" or STATE2KX="51" or STATE2KX="54" or
   STATE2KX="01" or STATE2KX="21" or STATE2KX="28" or STATE2KX="47" or
   STATE2KX="05" or STATE2KX="22" or STATE2KX="40" or STATE2KX="48"
 then region="3. South";
if STATE2KX="04" or STATE2KX="08" or STATE2KX="16" or STATE2KX="30" or STATE2KX="32" or
   STATE2KX="35" or STATE2KX="49" or STATE2KX="56" or STATE2KX="02" or
   STATE2KX="06" or STATE2KX="15" or STATE2KX="41" or STATE2KX="53"
 then region="4. West";

 /*CREATE VARIABLE FOR CENSUS DIVISION*/
length division $14.;
if STATE2K="09" or STATE2K="23" or STATE2K="25" or STATE2K="33" or STATE2K="44" or STATE2K="50"
    then division="1. New England";
if STATE2K="34" or STATE2K="36" or STATE2K="42"
    then division="2. Mid Atlantic";
if STATE2K="17" or STATE2K="18" or STATE2K="26" or STATE2K="39" or STATE2K="55"
  then division="3. E N Central";
if STATE2K="19" or STATE2K="20" or STATE2K="27" or STATE2K="29" or STATE2K="31" or STATE2K="38" or STATE2K="46"
  then division="4. W N Central";
if STATE2K="10" or STATE2K="11" or STATE2K="12" or STATE2K="13" or STATE2K="24"
   or STATE2K="37" or STATE2K="45" or STATE2K="51" or STATE2K="54"
   then division="5. South Atlantic";
if STATE2K="01" or STATE2K="21" or STATE2K="28" or STATE2K="47"
   then division="6. E S Central";
if STATE2K="05" or STATE2K="22" or STATE2K="40" or STATE2K="48"
   then division="7. W S Central";
if STATE2K="04" or STATE2K="08" or STATE2K="16" or STATE2K="30" or STATE2K="32" or STATE2K="35" or STATE2K="49"
   then division="8. Mountain";
if STATE2K="06" or STATE2K="15" or STATE2K="41" or STATE2K="53"
   then division="9. Pacific";
*Repeat with 2KX;
if STATE2KX="09" or STATE2KX="23" or STATE2KX="25" or STATE2KX="33" or STATE2KX="44" or STATE2KX="50"
    then division="1. New England";
if STATE2KX="34" or STATE2KX="36" or STATE2KX="42"
    then division="2. Mid Atlantic";
if STATE2KX="17" or STATE2KX="18" or STATE2KX="26" or STATE2KX="39" or STATE2KX="55"
  then division="3. E N Central";
if STATE2KX="19" or STATE2KX="20" or STATE2KX="27" or STATE2KX="29" or STATE2KX="31" or STATE2KX="38" or STATE2KX="46"
```

```
  then division="4. W N Central";
 if STATE2KX="10" or STATE2KX="11" or STATE2KX="12" or STATE2KX="13" or STATE2KX="24"
    or STATE2KX="37" or STATE2KX="45" or STATE2KX="51" or STATE2KX="54"
    then division="5. South Atlantic";
 if STATE2KX="01" or STATE2KX="21" or STATE2KX="28" or STATE2KX="47"
    then division="6. E S Central";
 if STATE2KX="05" or STATE2KX="22" or STATE2KX="40" or STATE2KX="48"
    then division="7. W S Central";
 if STATE2KX="04" or STATE2KX="08" or STATE2KX="16" or STATE2KX="30" or STATE2KX="32" or STATE2KX="35" or STATE2KX="49"
    then division="8. Mountain";
 if STATE2KX="06" or STATE2KX="15" or STATE2KX="41" or STATE2KX="53"
    then division="9. Pacific";

 /*CREATE VARIABLE FOR SURVEY YEAR BUCKETS*/
 length survey_year_buckets $12.;
 if survey_year >= 1986 and survey_year <= 1998 then survey_year_buckets="1. 1986-1998";
 if survey_year >= 1999 and survey_year <= 2015 then survey_year_buckets="2. 1999-2015";

 /*CREATE VARIABLE FOR RACE-ETHNICITY*/
 length race_eth $15.;
 race_eth = "Other/Unknown";
 if hisp = "Yes" then race_eth = "Hispanic";
 if hisp = "No" and race = "White" then race_eth = "White non-hisp";
 if hisp = "No" and race = "Black" then race_eth = "Black non-hisp";

 /*Assign Income Buckets for 1995 and 1996*/
 length income_buckets $20.;
 if income^=. and income < 35000 then income_buckets="1. Under $35,000";
 if income >= 35000 and income < 50000 then income_buckets="2. $35,000 - $50,000";
 if income >= 50000 and income < 75000 then income_buckets="3. $50,000 - $75,000";
 if income >=75000 then income_buckets="4. $75,000 and over";

 /*DROP INDIVIDUALS WITH MISSING DATES OF INTERVIEW OR DEATH*/
 if INTVDAY = 99 or INTVMONTH = 99 or DODDAY = 99 or DODMONTH = 99 then delete;

 /*CREATE VARIABLES FOR SURVIVAL TIME*/
 *Generate SAS date for interview date;
 intv_date =.;
 format intv_date DATE9.;
 intv_date = mdy(INTVMONTH, INTVDAY, INTVYEAR);
 *Date of Death;
 death_date =.;
 format death_date DATE9.;
 death_date = mdy(DODMONTH, DODDAY, DODYEAR);
 *Followup date, which we assume to be the last day of 2015;
 followup_date = mdy(12, 31, 2015);
 format death_date DATE9.;
 *Generate variable for survival time;
 survival_time =.;
 if MORTSTAT = 0 and intv_date^=. then survival_time = followup_date - intv_date; /*If assumed alive*/
 if MORTSTAT = 1 and death_date^=. and intv_date^=.
  then survival_time = death_date - intv_date; /*If assumed deceased*/

 /*CREATE VARIABLES FOR COHORT STATUS*/
 *Our full cohort includes individuals with data for pollution, urban/rural age, sex, race, marital status, income, and education;
 full_cohort = 0;
 if age >=18 and age <=84 and age ^=. and male ^=. and race_eth ^="" and marital_status ^="" and income_buckets ^=""
    and school_buckets ^="" and urban_rural ^="" and region ^="" and survey_year ^=. and pm ^=. and survival_time^=.
    then full_cohort = 1;
 *Our subcohort includes those individuals from the full cohort that have also have data on BMI and smoking status;
 subcohort = 0;
 if full_cohort = 1 and bmi_buckets ^= "" and smoking_status ^= ""
    then subcohort = 1;
run;
```

```
data FINAL;
 set FINAL;
 *Generate variables for specific-cause mortality. ICD-10 codes are includes in parentheses.
 *Diseases of heart (I00-I09, I11, I13, I20-I51) **INCLUDES UCOD 065-068**;
 heart_mort=0;
 if UCOD_113 = "054" or UCOD_113 = "055" or UCOD_113 = "056" or UCOD_113 = "057" or UCOD_113 = "058" or UCOD_113 = "059"
    or UCOD_113 = "060" or UCOD_113 = "061" or UCOD_113 = "062" or UCOD_113 = "063" or UCOD_113 = "064"
    or UCOD_113 = "065" or UCOD_113 = "066" or UCOD_113 = "067" or UCOD_113 = "068"
    then heart_mort = 1;
 *Cerebrovascular diseases (I60-I69);
 stroke_mort=0;
 if UCOD_113 = "070"
    then stroke_mort = 1;
 *Chronic lower respiratory diseases (J40-J47);
 resp_mort=0;
 if UCOD_113 = "082" or UCOD_113 = "083" or UCOD_113 = "084" or UCOD_113 = "085" or UCOD_113 = "086"
    then resp_mort = 1;
 *Influenza and pneumonia (J09-J18);
 flu_mort=0;
 if UCOD_113 = "076" or UCOD_113 = "077" or UCOD_113 = "078"
    then flu_mort = 1;

 *General cardiopulmonary disease (includes cardiovascular, cerebrovascular, respiratory, and influenza mortality);
 cardiopulm_mort=0;
 if heart_mort = 1 or stroke_mort = 1 or resp_mort = 1 or flu_mort = 1
    then cardiopulm_mort = 1;
run;

/*Check means*/
proc means data=work.FINAL;
 var pm pm25_99to15 survival_time age WGT_NEW;
run;

/*Check freqs*/
proc freq data=work.FINAL;
  tables urban_rural region age_groups survey_year survey_year_buckets age_buckets
   hisp race race_eth full_cohort subcohort male marital_status income_buckets school_buckets;
run;
proc freq data=FINAL;
 tables MORTSTAT heart_mort stroke_mort resp_mort flu_mort cancer_mort other_mort lungcancer_mort
   othercancer_mort cardiopulm_mort cause_of_death cancer_type;
run;
proc freq data=FINAL;
 tables MORTSTAT*survey_year;
run;
proc freq data=FINAL;
 tables full_cohort subcohort;
run;


/*Create full and subcohort datasets*/
data full_cohort;
 set FINAL;
 if full_cohort ^= 1 then delete;
run;

data sub_cohort;
 set full_cohort;
 if subcohort ^= 1 then delete;
 rename region=nregion;
run;
```

```
/*Create working dataset for CPH IPW analysis (subcohort only)*/
data dataset;
 set sub_cohort;

 divorced = 0;
 separated = 0;
 nevermarried = 0;
 widowed = 0;
 married = 0;
 if marital_status="1. Divorced" then divorced = 1;
 if marital_status="2. Separated" then separated = 1;
 if marital_status="3. Never marrie" then nevermarried = 1;
 if marital_status="4. Widowed" then widowed = 1;
 if marital_status="5. Married" then married = 1;

 hispanic = 0;
 black = 0;
 white_r = 0;
 asian = 0;
 if hisp="Yes" then hispanic = 1;
 if race="Black" then black = 1;
 if race="White" then white_r = 1;
 if race="Asian/Pacific Islander" then asian = 1;

 income1 = 0;
 income2 = 0;
 income3 = 0;
 income4 = 0;
 if income_buckets = "1. Under $35,000" then income1 = 1;
 if income_buckets = "2. $35,000 - $50,000" then income2 = 1;
 if income_buckets = "3. $50,000 - $75,000" then income3 = 1;
 if income_buckets = "4. $75,000 and over" then income4 = 1;

 bmi1 = 0;
 bmi2 = 0;
 bmi3 = 0;
 bmi4 = 0;
 bmi5 = 0;
 if bmi_buckets = "1. Below 20" then bmi1 = 1;
 if bmi_buckets = "2. 25-30" then bmi2 = 1;
 if bmi_buckets = "3. 30-35" then bmi3 = 1;
 if bmi_buckets = "4. 35 and Above" then bmi4 = 1;
 if bmi_buckets = "5. 20 - 25" then bmi5 = 1;

 school1 = 0;
 school2 = 0;
 school3 = 0;
 school4 = 0;
 school5 = 0;
 if school_buckets = "1. High-school Graduate" then school1 = 1;
 if school_buckets = "2. Some College" then school2 = 1;
 if school_buckets = "3. College Graduate" then school3 = 1;
 if school_buckets = "4. Post-College Graduate" then school4 = 1;
 if school_buckets = "5. <High-school Graduate" then school5 = 1;

 smok_current = 0;
 smok_former = 0;
 smok_never = 0;
 if smoking_status = "Current" then smok_current = 1;
 if smoking_status = "Former" then smok_former = 1;
 if smoking_status = "Never" then smok_never = 1;

 region1 = 0;
 region2 = 0;
```

```
 region3 = 0;
 region4 = 0;
 if nregion="1. Northeast" then region1 = 1;
 if nregion="2. Midwest" then region2 = 1;
 if nregion="3. South" then region3 = 1;
 if nregion="4. West" then region4 = 1;

 counter = 1;
run;


*************************************************************************************************;
* Weight Construction and Balance Assessment;
* Code adapted from Naimi et al. 2014;
*************************************************************************************************;
* NORMAL HOMOSCEDASTIC;
* Denominator;
proc genmod data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural survey_year ;
 model pm = age male race_eth smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_y
ear  / maxiter=100;
 *variance of pm is sse/df = valuedf;
 ods output Modelfit=ss(where=(criterion="Deviance"));
 output out=x xbeta=xb;*predicted mean is xb;
run;
* Match dataset with valuedf (variance of pm);
data ss;
 set ss;
 merg=1;
run;
data x;
 set x;
 merg=1;
run;
* Generate denominator density values;
data dataset;
  merge x ss;
  by merg;
  sd_pm = sqrt(valuedf); *standard deviation of pm;
  pdf_den_denom = pdf("normal",pm,xb,sd_pm);*denominator density value;

  drop criterion df value valuedf xb sd_pm;
run;

* Numerator;
proc genmod data=dataset;
 model pm = ;
 *variance of pm is sse/df = valuedf;
 ods output Modelfit=ss(where=(criterion="Deviance"));
 output out=x xbeta=xb;*predicted mean is xb;
run;
* Match dataset with valuedf (variance of pm);
data ss;
 set ss;
 merg=1;
run;
data x;
 set x;
 merg=1;
run;
* Generate numerator density values;
data dataset;
  merge x ss;
  by merg;
```

```
  sd_pm = sqrt(valuedf); *standard deviation of pm;
  pdf_den_numer = pdf("normal",pm,xb,sd_pm); *numerator density value;

  drop criterion df value valuedf;

  * Creating weights;
  sw_norm_hom = pdf_den_numer / pdf_den_denom;

  drop pdf_den_numer pdf_den_denom xb sd_pm;
run;


* t, CONSTANT VARIANCE;
* 1 df;
* Denominator;
proc genmod data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural survey_year ;
 model pm = age male race_eth smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_y
ear  / maxiter=1OO;
 output out=x stdresdev=e;
run;
data dataset;
  set x;
  pdf_den_denom1 = pdf("t", e, 1);
  *denominator density value with 1 df;

  drop e;
run;
*Numerator;
proc genmod data=dataset;
 model pm = ;
 output out=x stdresdev=e;
run;
data dataset;
  set x;
  pdf_den_numer1 = pdf("t", e, 1);
  *denominator density value with 1 df;

  * creating weights;
  sw_t_hom1 = pdf_den_numer1 / pdf_den_denom1;
  drop pdf_den_numer1 pdf_den_denom1 e;
run;

* 5 dfs;
* Denominator;
proc genmod data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural survey_year ;
 model pm = age male race_eth smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_y
ear  / maxiter=1OO;
 output out=x stdresdev=e;
run;
data dataset;
  set x;
  pdf_den_denom5 = pdf("t", e, 5);
  *denominator density value with 1 df;

  drop e;
run;
*Numerator;
proc genmod data=dataset;
 model pm = ;
 output out=x stdresdev=e;
run;
data dataset;
```

```
  set x;
  pdf_den_numer5 = pdf("t", e, 5);
  *denominator density value with 1 df;

  * creating weights;
  sw_t_hom5 = pdf_den_numer5 / pdf_den_denom5;
  drop pdf_den_numer5 pdf_den_denom5 e;
run;



* GAMMA;
* Denominator;
proc genmod data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural survey_year ;
 model pm = age male race_eth smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_y
ear  / dist=gamma link=log maxiter=100;
 ods output ParameterEstimates=s(where=(Parameter="Scale"));
 output out=x p=xb;
data s;
 set s;
 merg=1;
 rename estimate=scale;
 drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;
run;
data x;
 set x;
 merg=1;
run;
data dataset;
 merge x s;
 by merg;
   *determine Gamma shape parameter;
 lambda = xb/scale;
   *estimate denominator density;
 pdf_den_denom = pdf("gamma",pm,scale,lambda);

 drop scale lambda xb;
run;
*Numerator;
proc genmod data=dataset;
 model pm = / dist=gamma link=log maxiter=100;
 ods output ParameterEstimates=s(where=(Parameter="Scale"));
 output out=x p=xb;
data s;
 set s;
 merg=1;
 rename estimate=scale;
 drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;
run;
data x;
 set x;
 merg=1;
run;
data dataset;
 merge x s;
 by merg;
   *determine Gamma shape parameter;
 lambda = xb/scale;
   *estimate denominator density;
 pdf_den_numer = pdf("gamma",pm,scale,lambda);

 * generate ipws;
 sw_gamma = pdf_den_numer / pdf_den_denom;
```

```
 drop pdf_den_numer pdf_den_denom scale lambda xb;
run;



* QUANTILE BINNING;
* 10 bins;
proc rank data=dataset out=x ties=low groups=10;
 var pm;
 ranks pm_r10bins;
run;

data x;
 set x;
 pm_r10bins = pm_r10bins + 1;
run;

proc logistic data=x desc;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural survey_year ;
 model pm_r10bins = age male race_eth smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / maxiter = 100;
 output out=dataset predprobs=i;
run;
quit;run;

data dataset;
 set dataset;
 array ii(*) ip_1-ip_10;
  *ip_1-ip_10 are predicted probabilities of being in category 1 to 10;

 do j=1 to 10;
  *set density to predicted probability of observed exposure category;
  if pm_r10bins = j then pdf_den10bins=ii(j);
 end;
 drop j ip_1-ip_10;
 sw_bin10 = 1/10/pdf_den10bins;
run;
quit;run;

* 20 bins;
proc rank data=dataset out=x ties=low groups=20;
 var pm;
 ranks pm_r20bins;
run;

data x;
 set x;
 pm_r20bins = pm_r20bins + 1;
run;

proc logistic data=x desc;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural survey_year ;
 model pm_r20bins = age male race_eth smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / maxiter = 100;
 output out=dataset predprobs=i;
run;
quit;run;

data dataset;
 set dataset;
 array ii(*) ip_1-ip_20;
  *ip_1-ip_20 are predicted probabilities of being in category 1 to 20;
```

```
 do j=1 to 20;
  *set density to predicted probability of observed exposure category;
  if pm_r20bins = j then pdf_den20bins=ii(j);
 end;
 drop j ip_1-ip_20;
 sw_bin20 = (1/20)/pdf_den20bins;
run;
quit;run;



* Dataset management;
proc datasets library=work;
 delete s ss x;
run;
quit;
run;

data dataset;
 set dataset;
 drop _FROM_ _FROM_2 _FROM_3 _FROM_4 _INTO_ _INTO_2 _INTO_3 _INTO_4 pm_r10bins pdf_den10bins
  pm_r20bins pdf_den20bins Level1;
run;
quit;run;
* End dataset management;



* Display PM distribution;
ods trace on;
ods pdf file="Z:\Output for Review\PM Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   var pm;
   histogram / midpoints=0.2 to 1.8 by 0.02
               lognormal
               gamma
      normal
               odstitle = title
      name = "histogram";
run;
ods pdf close;



* Weight analysis - summary statistics and histograms of estimated weights;
*Normal, homoskedastic;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH Norm Hom Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_norm_hom / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_norm_hom;
 output out=temp.means_norm_hom mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=p
erc_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_norm_hom outfile="Z:\Output for Review\CPH Norm Hom Weights - Means.csv" dbms=csv replace;
run;
```

```
*T, 1;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH T1 Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_t_hom1 / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_t_hom1;
 output out=temp.means_t_hom1 mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=per
c_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_t_hom1 outfile="Z:\Output for Review\CPH T1 Weights - Means.csv" dbms=csv replace;
run;

*T, 5;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH T5 Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_t_hom5 / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_t_hom5;
 output out=temp.means_t_hom5 mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=per
c_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_t_hom5 outfile="Z:\Output for Review\CPH T5 Weights - Means.csv" dbms=csv replace;
run;

*Gamma;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH Gamma Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_gamma / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_gamma;
 output out=temp.means_gamma mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=perc
_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_gamma outfile="Z:\Output for Review\CPH Gamma Weights - Means.csv" dbms=csv replace;
run;

*Bin10;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH Bin10 Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_bin10 / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_bin10;
 output out=temp.means_bin10 mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=perc
```

```
_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_bin10 outfile="Z:\Output for Review\CPH Bin 10 Weights - Means.csv" dbms=csv replace;
run;

*Bin20;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH Bin20 Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_bin20 / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_bin20;
 output out=temp.means_bin20 mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=perc
_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_bin20 outfile="Z:\Output for Review\CPH Bin 20 Weights - Means.csv" dbms=csv replace;
run;


* Linear Analysis - Find R-squared of covariates on pm to check balance;
* No Weights;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 ods output FitStatistics=temp.fit_control ParameterEstimates=temp.param_control;
run;quit;run;
proc export data= temp.fit_control outfile="Z:\Output for Review\CPH Linear Fit - Control.csv" dbms=csv replace;
run;
proc export data= temp.param_control outfile="Z:\Output for Review\CPH Linear Params - Control.csv" dbms=csv replace;
run;

* Normal Hom;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_norm_hom;
 ods output fitstatistics=temp.fit_norm_hom parameterestimates=temp.param_norm_hom;
run;quit;run;
proc export data= temp.fit_norm_hom outfile="Z:\Output for Review\CPH Linear Fit - Norm Hom.csv" dbms=csv replace;
run;
proc export data= temp.param_norm_hom outfile="Z:\Output for Review\CPH Linear Params - Norm Hom.csv" dbms=csv replace;
run;

* T 1;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_t_hom1;
 ods output fitstatistics=temp.fit_t_hom1 parameterestimates=temp.param_t_hom1;
run;quit;run;
proc export data= temp.fit_t_hom1 outfile="Z:\Output for Review\CPH Linear Fit - T1.csv" dbms=csv replace;
run;
proc export data= temp.param_t_hom1 outfile="Z:\Output for Review\CPH Linear Params - T1.csv" dbms=csv replace;
run;

* T 5;
proc glm data=dataset;
```

```
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_t_hom5;
 ods output fitstatistics=temp.fit_t_hom5 parameterestimates=temp.param_t_hom5;
run;quit;run;
proc export data= temp.fit_t_hom5 outfile="Z:\Output for Review\CPH Linear Fit - T5.csv" dbms=csv replace;
run;
proc export data= temp.param_t_hom5 outfile="Z:\Output for Review\CPH Linear Params - T5.csv" dbms=csv replace;
run;

* Gamma;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_gamma;
 ods output fitstatistics=temp.fit_gamma parameterestimates=temp.param_gamma;
run;quit;run;
proc export data= temp.fit_gamma outfile="Z:\Output for Review\CPH Linear Fit - Gamma.csv" dbms=csv replace;
run;
proc export data= temp.param_gamma outfile="Z:\Output for Review\CPH Linear Params - Gamma.csv" dbms=csv replace;
run;

* Bin10;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_bin10;
 ods output fitstatistics=temp.fit_bin10 parameterestimates=temp.param_bin10;
run;quit;run;
proc export data= temp.fit_bin10 outfile="Z:\Output for Review\CPH Linear Fit - Bin10.csv" dbms=csv replace;
run;
proc export data= temp.param_bin10 outfile="Z:\Output for Review\CPH Linear Params - Bin10.csv" dbms=csv replace;
run;

* Bin20;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_bin20;
 ods output fitstatistics=temp.fit_bin20 parameterestimates=temp.param_bin20;
run;quit;run;
proc export data= temp.fit_bin20 outfile="Z:\Output for Review\CPH Linear Fit - Bin20.csv" dbms=csv replace;
run;
proc export data= temp.param_bin20 outfile="Z:\Output for Review\CPH Linear Params - Bin20.csv" dbms=csv replace;
run;

* Dataset management;
proc datasets library=temp;
 delete fit_control fit_norm_hom fit_t_hom1 fit_t_hom5 fit_gamma fit_bin10 fit_bin20
   means_norm_hom  means_t_hom1  means_t_hom5 means_gamma means_bin10  means_bin20
   param_control param_norm_hom  param_t_hom1  param_t_hom5 param_gamma param_bin10  param_bin20 ;
run;
quit;
run;
* End dataset management;


* Quantile Blocking;
proc means data = dataset p25 p50 p75;
 var pm;
```

```
 output out=testing_block_pm p25=pm_perc_25 p50=pm_perc_50 p75=pm_perc_75;
run;
data testing_block_pm;
 set testing_block_pm;
 merg_block=1;
run;
data dataset;
 set dataset;
 merg_block=1;
run;
data dataset;
 merge dataset testing_block_pm;
 by merg_block;
run;
* 1st block;
data dataset1yes;
 set dataset;
 if pm > pm_perc_25 then delete;
run;
data dataset1no;
 set dataset;
 if pm <= pm_perc_25 then delete;
run;
* 2nd block;
data dataset2yes;
 set dataset;
 if pm <= pm_perc_25 then delete;
 if pm > pm_perc_50 then delete;
run;
data dataset2no;
 set dataset;
 if pm <= pm_perc_50 and pm > pm_perc_25 then delete;
run;
* 3rd block;
data dataset3yes;
 set dataset;
 if pm <= pm_perc_50 then delete;
 if pm > pm_perc_75 then delete;
run;
data dataset3no;
 set dataset;
 if pm <= pm_perc_75 and pm > pm_perc_50 then delete;
run;
* 4th block;
data dataset4yes;
 set dataset;
 if pm <= pm_perc_75 then delete;
run;
data dataset4no;
 set dataset;
 if pm > pm_perc_75 then delete;
run;

* Macro for assessing quantile imbalance - without weights;
%macro block_test_nw(quantile);

    proc freq data = dataset&quantile.yes;
   tables counter / out = testing&quantile.yes;
    run;
    data testing&quantile.yes;
   set testing&quantile.yes;
   dataset&quantile.yes_n = count;
   drop percent count;
    run;
```

```
    proc freq data = dataset&quantile.no;
   tables counter / out = testing&quantile.no;
    run;
    data testing&quantile.no;
   set testing&quantile.no;
   dataset&quantile.no_n = count;
   drop percent count;
    run;
    data dataset&quantile._counts;
        merge testing&quantile.yes testing&quantile.no;
        by counter;
    run;

    * Standardized differences;
    proc means data=dataset&quantile.yes mean stddev stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_&quantile.yes;
    run;
 data test_means_&quantile.yes;
  set test_means_&quantile.yes;
  rename mean=average&quantile.yes stddev=std&quantile.yes;
  id = _N_;
 run;
    proc means data=dataset&quantile.no mean stddev stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_&quantile.no;
    run;
 data test_means_&quantile.no;
  set test_means_&quantile.no;
  rename mean=average&quantile.no stddev=std&quantile.no;
  id = _N_;
 run;
    data dataset&quantile.merged;
        merge test_means_&quantile.yes test_means_&quantile.no;
        by id;
  counter=1;
    run;

    * Merge all, create standardized difference statistics;
    data dataset&quantile._blocking;
        merge dataset&quantile.merged dataset&quantile._counts;
        by counter;
        standard_error=sqrt((std&quantile.yes**2)/dataset&quantile.yes_n + (std&quantile.no**2)/dataset&quantile.no_n);
        standardized_diff&quantile.=(average&quantile.yes - average&quantile.no)/standard_error;
        keep variable standardized_diff&quantile. id;
    run;

%mend;

%block_test_nw(1);
%block_test_nw(2);
%block_test_nw(3);
%block_test_nw(4);

data temp.quantile_blocking_nw(drop=id);
 merge dataset1_blocking dataset2_blocking dataset3_blocking dataset4_blocking;
 by id;
run;
```

```
proc export data= temp.quantile_blocking_nw outfile="Z:\Output for Review\CPH Control - Q Blocks.csv" dbms=csv replace;
run;


* Macro for assessing quantile imbalance - with weights;
%macro block_test_weights(ipw);
 * 1st quantile;
    proc freq data = dataset1yes;
   tables counter / out = testing1yes;
  weight &ipw.;
    run;
    data testing1yes;
   set testing1yes;
   dataset1yes_n = count;
   drop table count;
  counter=1;
    run;
    proc freq data = dataset1no;
   tables counter/ out = testing1no;
  weight &ipw.;
    run;
    data testing1no;
   set testing1no;
   dataset1no_n = count;
   drop table count;
  counter=1;
    run;
 data dataset1_counts;
        merge testing1yes testing1no;
        by counter;
    run;
    * Standardized differences;
    proc means data=dataset1yes mean std stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
                income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
                region1 region2 region3 region4 rural;
        ods output Summary=test_means_1yes;
  weight &ipw.;
    run;
 data test_means_1yes;
  set test_means_1yes;
  rename mean=average1yes stddev=std1yes;
  id = _N_;
 run;
    proc means data=dataset1no mean std stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
                income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
                region1 region2 region3 region4 rural;
        ods output Summary=test_means_1no;
  weight &ipw.;
    run;
 data test_means_1no;
  set test_means_1no;
  rename mean=average1no stddev=std1no;
  id = _N_;
 run;
    data dataset1merged;
        merge test_means_1yes test_means_1no;
        by id;
  counter=1;
    run;
    * Merge all, create standardized difference statistics;
```

```
    data dataset1_blocking;
        merge dataset1merged dataset1_counts;
        by counter;
        standard_error=sqrt((std1yes**2)/dataset1yes_n + (std1no**2)/dataset1no_n);
        standardized_diff1=(average1yes - average1no)/standard_error;
        keep variable standardized_diff1 id;
    run;

 * 2nd Quantile;
    proc freq data = dataset2yes;
   tables counter / out = testing2yes;
  weight &ipw.;
    run;
    data testing2yes;
   set testing2yes;
   dataset2yes_n = count;
   drop table count;
  counter=1;
    run;
    proc freq data = dataset2no;
   tables counter / out = testing2no;
  weight &ipw.;
    run;
    data testing2no;
   set testing2no;
   dataset2no_n = count;
   drop table count;
  counter=1;
    run;
 data dataset2_counts;
        merge testing2yes testing2no;
        by counter;
    run;
    * Standardized differences;
    proc means data=dataset2yes mean std stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_2yes;
  weight &ipw.;
    run;
 data test_means_2yes;
  set test_means_2yes;
  rename mean=average2yes stddev=std2yes;
  id = _N_;
 run;
    proc means data=dataset2no mean std stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_2no;
  weight &ipw.;
    run;
 data test_means_2no;
  set test_means_2no;
  rename mean=average2no stddev=std2no;
  id = _N_;
 run;
    data dataset2merged;
        merge test_means_2yes test_means_2no;
        by id;
  counter=1;
```

```
    run;
    * Merge all, create standardized difference statistics;
    data dataset2_blocking;
        merge dataset2merged dataset2_counts;
        by counter;
        standard_error=sqrt((std2yes**2)/dataset2yes_n + (std2no**2)/dataset2no_n);
        standardized_diff2=(average2yes - average2no)/standard_error;
        keep variable standardized_diff2 id;
    run;

 * 3rd Quantile;
    proc freq data = dataset3yes;
   tables counter / out = testing3yes;
  weight &ipw.;
    run;
    data testing3yes;
   set testing3yes;
   dataset3yes_n = count;
   drop table count;
  counter=1;
    run;
    proc freq data = dataset3no;
   tables counter / out = testing3no;
  weight &ipw.;
    run;
    data testing3no;
   set testing3no;
   dataset3no_n = count;
   drop table count;
  counter=1;
    run;
 data dataset3_counts;
        merge testing3yes testing3no;
        by counter;
    run;
    * Standardized differences;
    proc means data=dataset3yes mean std stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_3yes;
  weight &ipw.;
    run;
 data test_means_3yes;
  set test_means_3yes;
  rename mean=average3yes stddev=std3yes;
  id = _N_;
 run;
    proc means data=dataset3no mean std stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_3no;
  weight &ipw.;
    run;
 data test_means_3no;
  set test_means_3no;
  rename mean=average3no stddev=std3no;
  id = _N_;
 run;
    data dataset3merged;
        merge test_means_3yes test_means_3no;
```

```
        by id;
 counter=1;
   run;
   * Merge all, create standardized difference statistics;
   data dataset3_blocking;
       merge dataset3merged dataset3_counts;
       by counter;
       standard_error=sqrt((std3yes**2)/dataset3yes_n + (std3no**2)/dataset3no_n);
       standardized_diff3=(average3yes - average3no)/standard_error;
       keep variable standardized_diff3 id;
   run;

* 4th Quantile;
   proc freq data = dataset4yes;
  tables counter / out = testing4yes;
 weight &ipw.;
   run;
   data testing4yes;
  set testing4yes;
  dataset4yes_n = count;
  drop table count;
 counter=1;
   run;
   proc freq data = dataset4no;
  tables counter / out = testing4no;
 weight &ipw.;
   run;
   data testing4no;
  set testing4no;
  dataset4no_n = count;
  drop table count;
 counter=1;
   run;
data dataset4_counts;
       merge testing4yes testing4no;
       by counter;
   run;
   * Standardized differences;
   proc means data=dataset4yes mean std stackodsoutput;
       var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
       ods output Summary=test_means_4yes;
 weight &ipw.;
   run;
data test_means_4yes;
 set test_means_4yes;
 rename mean=average4yes stddev=std4yes;
 id = _N_;
run;
   proc means data=dataset4no mean std stackodsoutput;
       var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
       ods output Summary=test_means_4no;
 weight &ipw.;
   run;
data test_means_4no;
 set test_means_4no;
 rename mean=average4no stddev=std4no;
 id = _N_;
run;
```

```
    data dataset4merged;
        merge test_means_4yes test_means_4no;
        by id;
  counter=1;
    run;
    * Merge all, create standardized difference statistics;
    data dataset4_blocking;
        merge dataset4merged dataset4_counts;
        by counter;
        standard_error=sqrt((std4yes**2)/dataset4yes_n + (std4no**2)/dataset4no_n);
        standardized_diff4=(average4yes - average4no)/standard_error;
        keep variable standardized_diff4 id;
    run;

* Merge all datasets;
data temp.qb_ipw_&ipw.(drop=id);
 merge dataset1_blocking dataset2_blocking dataset3_blocking dataset4_blocking;
 by id;
run;

proc export data= temp.qb_ipw_&ipw. outfile="Z:\Output for Review\CPH &ipw. - Q Blocks.csv" dbms=csv replace;
run;

%mend block_test_weights;


%block_test_weights(sw_norm_hom);
%block_test_weights(sw_t_hom1);
%block_test_weights(sw_t_hom5);
%block_test_weights(sw_gamma);
%block_test_weights(sw_bin10);
%block_test_weights(sw_bin20);


* Truncated weights at 1st and 99th percentiles - using percentiles of estimated weights;
* Precise values were used during the analysis, which are here ommitted due to the restricted-use nature of the data;
data dataset;
 set dataset;
 sw_norm_hom_T = sw_norm_hom;
 if sw_norm_hom < ####### then sw_norm_hom_T = #######;
 if sw_norm_hom > ####### then sw_norm_hom_T = #######;

 sw_t_hom1_T = sw_t_hom1;
 if sw_t_hom1 < ####### then sw_t_hom1_T = #######;
 if sw_t_hom1 > ####### then sw_t_hom1_T = #######;

 sw_t_hom5_T = sw_t_hom5;
 if sw_t_hom5 < ####### then sw_t_hom5_T = #######;
 if sw_t_hom5 > ####### then sw_t_hom5_T = #######;

 sw_gamma_T = sw_gamma;
 if sw_gamma < ####### then sw_gamma_T = #######;
 if sw_gamma > ####### then sw_gamma_T = #######;

 sw_bin10_T = sw_bin10;
 if sw_bin10 < ####### then sw_bin10_T = #######;
 if sw_bin10 > ####### then sw_bin10_T = #######;

 sw_bin20_T = sw_bin20;
 if sw_bin20 < ####### then sw_bin20_T = #######;
 if sw_bin20 > ####### then sw_bin20_T = #######;
run;
```

```
* Truncated Weight Analysis;
*Normal, homoskedastic;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH Norm Hom_T Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_norm_hom_T / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_norm_hom_T;
 output out=temp.means_norm_hom_T mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50
=perc_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_norm_hom_T outfile="Z:\Output for Review\CPH Norm Hom_T Weights - Means.csv" dbms=csv replace;
run;

*T, 1;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH T1_T Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_t_hom1_T / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_t_hom1_T;
 output out=temp.means_t_hom1_T mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=p
erc_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_t_hom1_T outfile="Z:\Output for Review\CPH T1_T Weights - Means.csv" dbms=csv replace;
run;

*T, 5;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH T5_T Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_t_hom5_T / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_t_hom5_T;
 output out=temp.means_t_hom5_T mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=p
erc_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_t_hom5_T outfile="Z:\Output for Review\CPH T5_T Weights - Means.csv" dbms=csv replace;
run;

*Gamma;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH Gamma_T Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_gamma_T / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_gamma_T;
 output out=temp.means_gamma_T mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=pe
```

```
rc_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_gamma_T outfile="Z:\Output for Review\CPH Gamma_T Weights - Means.csv" dbms=csv replace;
run;

*Bin10;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH Bin10_T Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_bin10_T / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_bin10_T;
 output out=temp.means_bin10_T mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=pe
rc_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_bin10_T outfile="Z:\Output for Review\CPH Bin 10_T Weights - Means.csv" dbms=csv replace;
run;

*Bin20;
ods graphics on;
ods trace on;
ods pdf file="Z:\Output for Review\CPH Bin20_T Distribution.pdf";
ods select histogram;
proc univariate data=dataset noprint;
   histogram sw_bin20_T / name = "histogram";
run;
ods pdf close;
proc means data = dataset mean min max range p1 p5 p10 p25 p50 p75 p90 p95 p99;
 var sw_bin20_T;
 output out=temp.means_bin20_T mean=average min=minimum max=maximum range=range p1=perc_1 p5=perc_5 p10=perc_10 p25=perc_25 p50=pe
rc_50 p75=perc_75 p90=perc_90 p95=perc_95 p99=perc_99;
run;
proc export data= temp.means_bin20_T outfile="Z:\Output for Review\CPH Bin 20_T Weights - Means.csv" dbms=csv replace;
run;

* Dataset management;
proc datasets library=temp;
 delete means_bin20 means_bin10 means_gamma means_t_hom5 means_t_hom1 means_norm_hom
  means_bin20_T means_bin10_T means_gamma_T means_t_hom5_T means_t_hom1_T means_norm_hom_T;
run;
quit;
run;
* End dataset management;


* Linear Analysis - As before, with truncated weights this time;
* Normal Hom;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_norm_hom_T;
 ods output fitstatistics=temp.fit_norm_hom parameterestimates=temp.param_norm_hom;
run;quit;run;
proc export data= temp.fit_norm_hom outfile="Z:\Output for Review\CPH Linear Fit - T Norm Hom.csv" dbms=csv replace;
run;
proc export data= temp.param_norm_hom outfile="Z:\Output for Review\CPH Linear Params - T Norm Hom.csv" dbms=csv replace;
run;

* T 1;
```

```
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_t_hom1_T;
 ods output fitstatistics=temp.fit_t_hom1 parameterestimates=temp.param_t_hom1;
run;quit;run;
proc export data= temp.fit_t_hom1 outfile="Z:\Output for Review\CPH Linear Fit - T T1.csv" dbms=csv replace;
run;
proc export data= temp.param_t_hom1 outfile="Z:\Output for Review\CPH Linear Params - T T1.csv" dbms=csv replace;
run;

* T 5;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_t_hom5_T;
 ods output fitstatistics=temp.fit_t_hom5 parameterestimates=temp.param_t_hom5;
run;quit;run;
proc export data= temp.fit_t_hom5 outfile="Z:\Output for Review\CPH Linear Fit - T T5.csv" dbms=csv replace;
run;
proc export data= temp.param_t_hom5 outfile="Z:\Output for Review\CPH Linear Params - T T5.csv" dbms=csv replace;
run;

* Gamma;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_gamma_T;
 ods output fitstatistics=temp.fit_gamma parameterestimates=temp.param_gamma;
run;quit;run;
proc export data= temp.fit_gamma outfile="Z:\Output for Review\CPH Linear Fit - T Gamma.csv" dbms=csv replace;
run;
proc export data= temp.param_gamma outfile="Z:\Output for Review\CPH Linear Params - T Gamma.csv" dbms=csv replace;
run;

* Bin1O;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_bin1O_T;
 ods output fitstatistics=temp.fit_bin1O parameterestimates=temp.param_bin1O;
run;quit;run;
proc export data= temp.fit_bin1O outfile="Z:\Output for Review\CPH Linear Fit - T Bin1O.csv" dbms=csv replace;
run;
proc export data= temp.param_bin1O outfile="Z:\Output for Review\CPH Linear Params - T Bin1O.csv" dbms=csv replace;
run;

* Bin2O;
proc glm data=dataset;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race hisp urban_rural;
 model pm = age male race hisp smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / solution;
 weight sw_bin2O_T;
 ods output fitstatistics=temp.fit_bin2O parameterestimates=temp.param_bin2O;
run;quit;run;
proc export data= temp.fit_bin2O outfile="Z:\Output for Review\CPH Linear Fit - T Bin2O.csv" dbms=csv replace;
run;
proc export data= temp.param_bin2O outfile="Z:\Output for Review\CPH Linear Params - T Bin2O.csv" dbms=csv replace;
run;
```

```
* Dataset management;
proc datasets library=temp;
 delete fit_control fit_norm_hom fit_t_hom1 fit_t_hom5 fit_gamma fit_bin10 fit_bin20
   means_norm_hom  means_t_hom1  means_t_hom5 means_gamma means_bin10  means_bin20
   param_control param_norm_hom  param_t_hom1  param_t_hom5 param_gamma param_bin10  param_bin20 ;
run;
quit;
run;
* End dataset management;

* Quantile Blocking - now with truncated weights;
proc means data = dataset p25 p50 p75;
 var pm;
 output out=testing_block_pm p25=pm_perc_25 p50=pm_perc_50 p75=pm_perc_75;
run;
data testing_block_pm;
 set testing_block_pm;
 merg_block=1;
run;
data dataset;
 set dataset;
 merg_block=1;
run;
data dataset;
 merge dataset testing_block_pm;
 by merg_block;
run;
* 1st block;
data dataset1yes;
 set dataset;
 if pm > pm_perc_25 then delete;
run;
data dataset1no;
 set dataset;
 if pm <= pm_perc_25 then delete;
run;
* 2nd block;
data dataset2yes;
 set dataset;
 if pm <= pm_perc_25 then delete;
 if pm > pm_perc_50 then delete;
run;
data dataset2no;
 set dataset;
 if pm <= pm_perc_50 and pm > pm_perc_25 then delete;
run;
* 3rd block;
data dataset3yes;
 set dataset;
 if pm <= pm_perc_50 then delete;
 if pm > pm_perc_75 then delete;
run;
data dataset3no;
 set dataset;
 if pm <= pm_perc_75 and pm > pm_perc_50 then delete;
run;
* 4th block;
data dataset4yes;
 set dataset;
 if pm <= pm_perc_75 then delete;
run;
data dataset4no;
 set dataset;
 if pm > pm_perc_75 then delete;
```

```
run;

* Macro for assessing quantile imbalance;
%macro block_test_nw(quantile);

    proc freq data = dataset&quantile.yes;
   tables counter / out = testing&quantile.yes;
    run;
    data testing&quantile.yes;
   set testing&quantile.yes;
   dataset&quantile.yes_n = count;
   drop percent count;
    run;
    proc freq data = dataset&quantile.no;
   tables counter / out = testing&quantile.no;
    run;
    data testing&quantile.no;
   set testing&quantile.no;
   dataset&quantile.no_n = count;
   drop percent count;
    run;
    data dataset&quantile._counts;
        merge testing&quantile.yes testing&quantile.no;
        by counter;
 run;
    * Standardized differences;
    proc means data=dataset&quantile.yes mean stddev stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_&quantile.yes;
    run;
 data test_means_&quantile.yes;
  set test_means_&quantile.yes;
  rename mean=average&quantile.yes stddev=std&quantile.yes;
  id = _N_;
 run;
    proc means data=dataset&quantile.no mean stddev stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_&quantile.no;
    run;
 data test_means_&quantile.no;
  set test_means_&quantile.no;
  rename mean=average&quantile.no stddev=std&quantile.no;
  id = _N_;
 run;
    data dataset&quantile.merged;
        merge test_means_&quantile.yes test_means_&quantile.no;
        by id;
  counter=1;
    run;
    * Merge all, create standardized difference statistics;
    data dataset&quantile._blocking;
        merge dataset&quantile.merged dataset&quantile._counts;
        by counter;
        standard_error=sqrt((std&quantile.yes**2)/dataset&quantile.yes_n + (std&quantile.no**2)/dataset&quantile.no_n);
        standardized_diff&quantile.=(average&quantile.yes - average&quantile.no)/standard_error;
        keep variable standardized_diff&quantile. id;
    run;
```

```
%mend;

%block_test_nw(1);
%block_test_nw(2);
%block_test_nw(3);
%block_test_nw(4);

data temp.quantile_blocking_nw(drop=id);
 merge dataset1_blocking dataset2_blocking dataset3_blocking dataset4_blocking;
 by id;
run;
proc export data= temp.quantile_blocking_nw outfile="Z:\Output for Review\CPH Control T - Q Blocks.csv" dbms=csv replace;
run;


* Macro for assessing quantile imbalance - with truncated weights;
%macro block_test_weights(ipw);
 * 1st quantile;
    proc freq data = dataset1yes;
   tables counter / out = testing1yes;
  weight &ipw.;
    run;
    data testing1yes;
   set testing1yes;
   dataset1yes_n = count;
   drop table count;
  counter=1;
    run;
    proc freq data = dataset1no;
   tables counter/ out = testing1no;
  weight &ipw.;
    run;
    data testing1no;
   set testing1no;
   dataset1no_n = count;
   drop table count;
  counter=1;
    run;
 data dataset1_counts;
        merge testing1yes testing1no;
        by counter;
    run;
    * Standardized differences;
    proc means data=dataset1yes mean std stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_1yes;
  weight &ipw.;
    run;
 data test_means_1yes;
  set test_means_1yes;
  rename mean=average1yes stddev=std1yes;
  id = _N_;
 run;
    proc means data=dataset1no mean std stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_1no;
  weight &ipw.;
    run;
```

```
 data test_means_1no;
  set test_means_1no;
  rename mean=average1no stddev=std1no;
  id = _N_;
 run;
    data dataset1merged;
        merge test_means_1yes test_means_1no;
        by id;
  counter=1;
    run;
    * Merge all, create standardized difference statistics;
    data dataset1_blocking;
        merge dataset1merged dataset1_counts;
        by counter;
        standard_error=sqrt((std1yes**2)/dataset1yes_n + (std1no**2)/dataset1no_n);
        standardized_diff1=(average1yes - average1no)/standard_error;
        keep variable standardized_diff1 id;
    run;

 * 2nd Quantile;
    proc freq data = dataset2yes;
   tables counter / out = testing2yes;
  weight &ipw.;
    run;
    data testing2yes;
   set testing2yes;
   dataset2yes_n = count;
   drop table count;
  counter=1;
    run;
    proc freq data = dataset2no;
   tables counter / out = testing2no;
  weight &ipw.;
    run;
    data testing2no;
   set testing2no;
   dataset2no_n = count;
   drop table count;
  counter=1;
    run;
 data dataset2_counts;
        merge testing2yes testing2no;
        by counter;
    run;
    * Standardized differences;
    proc means data=dataset2yes mean std stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
                income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
                region1 region2 region3 region4 rural;
        ods output Summary=test_means_2yes;
  weight &ipw.;
    run;
 data test_means_2yes;
  set test_means_2yes;
  rename mean=average2yes stddev=std2yes;
  id = _N_;
 run;
    proc means data=dataset2no mean std stackodsoutput;
        var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
                income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
                region1 region2 region3 region4 rural;
        ods output Summary=test_means_2no;
```

```
 weight &ipw.;
   run;
data test_means_2no;
 set test_means_2no;
 rename mean=average2no stddev=std2no;
 id = _N_;
run;
   data dataset2merged;
       merge test_means_2yes test_means_2no;
       by id;
 counter=1;
   run;
   * Merge all, create standardized difference statistics;
   data dataset2_blocking;
       merge dataset2merged dataset2_counts;
       by counter;
       standard_error=sqrt((std2yes**2)/dataset2yes_n + (std2no**2)/dataset2no_n);
       standardized_diff2=(average2yes - average2no)/standard_error;
       keep variable standardized_diff2 id;
   run;

* 3rd Quantile;
   proc freq data = dataset3yes;
  tables counter / out = testing3yes;
 weight &ipw.;
   run;
   data testing3yes;
  set testing3yes;
  dataset3yes_n = count;
  drop table count;
 counter=1;
   run;
   proc freq data = dataset3no;
  tables counter / out = testing3no;
 weight &ipw.;
   run;
   data testing3no;
  set testing3no;
  dataset3no_n = count;
  drop table count;
 counter=1;
   run;
data dataset3_counts;
       merge testing3yes testing3no;
       by counter;
   run;
* Standardized differences;
   proc means data=dataset3yes mean std stackodsoutput;
       var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
       ods output Summary=test_means_3yes;
 weight &ipw.;
   run;
data test_means_3yes;
 set test_means_3yes;
 rename mean=average3yes stddev=std3yes;
 id = _N_;
run;
   proc means data=dataset3no mean std stackodsoutput;
       var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
```

```
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_3no;
 weight &ipw.;
   run;
data test_means_3no;
 set test_means_3no;
 rename mean=average3no stddev=std3no;
 id = _N_;
run;
   data dataset3merged;
       merge test_means_3yes test_means_3no;
       by id;
 counter=1;
   run;
   * Merge all, create standardized difference statistics;
   data dataset3_blocking;
       merge dataset3merged dataset3_counts;
       by counter;
       standard_error=sqrt((std3yes**2)/dataset3yes_n + (std3no**2)/dataset3no_n);
       standardized_diff3=(average3yes - average3no)/standard_error;
       keep variable standardized_diff3 id;
   run;

* 4th Quantile;
   proc freq data = dataset4yes;
  tables counter / out = testing4yes;
 weight &ipw.;
   run;
   data testing4yes;
  set testing4yes;
  dataset4yes_n = count;
  drop table count;
 counter=1;
   run;
   proc freq data = dataset4no;
  tables counter / out = testing4no;
 weight &ipw.;
   run;
   data testing4no;
  set testing4no;
  dataset4no_n = count;
  drop table count;
 counter=1;
   run;
data dataset4_counts;
       merge testing4yes testing4no;
       by counter;
   run;
   * Standardized differences;
   proc means data=dataset4yes mean std stackodsoutput;
       var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
       ods output Summary=test_means_4yes;
 weight &ipw.;
   run;
data test_means_4yes;
 set test_means_4yes;
 rename mean=average4yes stddev=std4yes;
 id = _N_;
run;
   proc means data=dataset4no mean std stackodsoutput;
       var age male  survey_year  divorced separated nevermarried widowed married hispanic black white_r asian income1 income2 in
```

```
come3
               income4 bmi1 bmi2 bmi3 bmi4 bmi5 school1 school2 school3 school4 school5 smok_current smok_former smok_never
               region1 region2 region3 region4 rural;
        ods output Summary=test_means_4no;
  weight &ipw.;
    run;
 data test_means_4no;
  set test_means_4no;
  rename mean=average4no stddev=std4no;
  id = _N_;
 run;
    data dataset4merged;
        merge test_means_4yes test_means_4no;
        by id;
  counter=1;
    run;
    * Merge all, create standardized difference statistics;
    data dataset4_blocking;
        merge dataset4merged dataset4_counts;
        by counter;
        standard_error=sqrt((std4yes**2)/dataset4yes_n + (std4no**2)/dataset4no_n);
        standardized_diff4=(average4yes - average4no)/standard_error;
        keep variable standardized_diff4 id;
    run;

* Merge all datasets;
data temp.qb_ipw_&ipw.(drop=id);
 merge dataset1_blocking dataset2_blocking dataset3_blocking dataset4_blocking;
 by id;
run;

proc export data= temp.qb_ipw_&ipw. outfile="Z:\Output for Review\CPH &ipw. - Q Blocks.csv" dbms=csv replace;
run;

%mend block_test_weights;


%block_test_weights(sw_norm_hom_T);
%block_test_weights(sw_t_hom1_T);
%block_test_weights(sw_t_hom5_T);
%block_test_weights(sw_gamma_T);
%block_test_weights(sw_bin1O_T);
%block_test_weights(sw_bin2O_T);



*************************************************************************************************;
* Analysis with Cox Proportional Hazards Models - delete "covs" options for naive standard errors;
*************************************************************************************************;
data temp.dataset;
 set dataset;
 keep pm age_buckets age male race_eth mortstat cardiopulm_mort survival_time
   counter sw_norm_hom sw_t_hom1 sw_t_hom5 sw_gamma sw_bin1O sw_bin2O
   sw_norm_hom sw_t_hom1 sw_t_hom5 sw_gamma sw_bin1O sw_bin2O
   smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_year;
run;

*ALL-CAUSE MORTALITY;
* MODELS WITH CONTROLS FOR COVARIATES;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
```

```
 weight counter;
 ods output ParameterEstimates=temp.ph_control;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_norm_hom;
 ods output ParameterEstimates=temp.ph_norm_hom;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_t_hom1;
 ods output ParameterEstimates=temp.ph_t_hom1;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_t_hom5;
 ods output ParameterEstimates=temp.ph_t_hom5;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_gamma;
 ods output ParameterEstimates=temp.ph_gamma;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_bin10;
 ods output ParameterEstimates=temp.ph_bin10;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_bin20;
 ods output ParameterEstimates=temp.ph_bin20;
run;

* Truncated weights;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_norm_hom_T;
 ods output ParameterEstimates=temp.ph_norm_hom_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
```

```
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_t_hom1_T;
 ods output ParameterEstimates=temp.ph_t_hom1_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_t_hom5_T;
 ods output ParameterEstimates=temp.ph_t_hom5_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_gamma_T;
 ods output ParameterEstimates=temp.ph_gamma_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_bin10_T;
 ods output ParameterEstimates=temp.ph_bin10_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 weight sw_bin20_T;
 ods output ParameterEstimates=temp.ph_bin20_T;
run;
* DRCPHM - Cox proportional hazards models with both IPW weighting (excluding the control case) and covariate inclusion in the mod
el,
abbreviated with "DR" to reference the doubly robust models to which they are most similar (though no doubly robust property has b
een
proven for these models - this is for organizational convenience);
proc export data= temp.ph_control outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - Control covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_norm_hom outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - Norm Hom covs.csv" dbms=csv replace
;
run;
proc export data= temp.ph_t_hom1 outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - T1 covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_t_hom5 outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - T5 covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_gamma outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - Gamma covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_bin10 outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - Bin10 covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_bin20 outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - Bin20 covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_norm_hom_T outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - T Norm Hom covs.csv" dbms=csv rep
lace;
run;
proc export data= temp.ph_t_hom1_T outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - T T1 covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_t_hom5_T outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - T T5 covs.csv" dbms=csv replace;
run;
```

```
proc export data= temp.ph_gamma_T outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - T Gamma covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_bin10_T outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - T Bin10 covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_bin20_T outfile="Z:\Output for Review\AC - Age Bins DRCPHM Output - T Bin20 covs.csv" dbms=csv replace;
run;
proc datasets library=temp;
 delete ph_control ph_norm_hom ph_t_hom1 ph_t_hom5 ph_gamma ph_bin10 ph_bin20 ph_norm_hom_T ph_t_hom1_T ph_t_hom5_T ph_gamma_T ph_
bin10_T ph_bin20_T;
run;
quit;
run;

* MSMs - referencing marginal structural models, without covariates included in the model;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight counter;
 ods output ParameterEstimates=temp.cmsm_control;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_norm_hom;
 ods output ParameterEstimates=temp.cmsm_norm_hom;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_t_hom1;
 ods output ParameterEstimates=temp.cmsm_t_hom1;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_t_hom5;
 ods output ParameterEstimates=temp.cmsm_t_hom5;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_gamma;
 ods output ParameterEstimates=temp.cmsm_gamma;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_bin10;
 ods output ParameterEstimates=temp.cmsm_bin10;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_bin20;
 ods output ParameterEstimates=temp.cmsm_bin20;
run;
```

```
* Truncated weights;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_norm_hom_T;
 ods output ParameterEstimates=temp.cmsm_norm_hom_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_t_hom1_T;
 ods output ParameterEstimates=temp.cmsm_t_hom1_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_t_hom5_T;
 ods output ParameterEstimates=temp.cmsm_t_hom5_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_gamma_T;
 ods output ParameterEstimates=temp.cmsm_gamma_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_bin10_T;
 ods output ParameterEstimates=temp.cmsm_bin10_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*mortstat(0) = pm / RL ;
 weight sw_bin20_T;
 ods output ParameterEstimates=temp.cmsm_bin20_T;
run;
proc export data= temp.cmsm_control outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - Control covs.csv" dbms=csv replace
;
run;
proc export data= temp.cmsm_norm_hom outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - Norm Hom covs.csv" dbms=csv repla
ce;
run;
proc export data= temp.cmsm_t_hom1 outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - T1 covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_t_hom5 outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - T5 covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_gamma outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - Gamma covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_bin10 outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - Bin10 covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_bin20 outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - Bin20 covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_norm_hom_T outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - T Norm Hom covs.csv" dbms=csv r
eplace;
run;
proc export data= temp.cmsm_t_hom1_T outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - T T1 covs.csv" dbms=csv replace;
run;
```

```
proc export data= temp.cmsm_t_hom5_T outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - T T5 covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_gamma_T outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - T Gamma covs.csv" dbms=csv replace
;
run;
proc export data= temp.cmsm_bin1O_T outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - T Bin1O covs.csv" dbms=csv replace
;
run;
proc export data= temp.cmsm_bin2O_T outfile="Z:\Output for Review\AC - Age Bins PH MSM Output - T Bin2O covs.csv" dbms=csv replace
;
run;
proc datasets library=temp;
 delete cmsm_control cmsm_norm_hom cmsm_t_hom1 cmsm_t_hom5 cmsm_gamma cmsm_bin1O cmsm_bin2O cmsm_norm_hom_T cmsm_t_hom1_T cmsm_t_h
om5_T
  cmsm_gamma_T cmsm_bin1O_T cmsm_bin2O_T;
run;
quit;
run;


*CARDIO-PULMONARY MORTALITY;
* MODELS WITH CONTROLS FOR COVARIATES;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight counter;
 ods output ParameterEstimates=temp.ph_control;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_norm_hom;
 ods output ParameterEstimates=temp.ph_norm_hom;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_t_hom1;
 ods output ParameterEstimates=temp.ph_t_hom1;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_t_hom5;
 ods output ParameterEstimates=temp.ph_t_hom5;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_gamma;
 ods output ParameterEstimates=temp.ph_gamma;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
```

```
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_bin10;
 ods output ParameterEstimates=temp.ph_bin10;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_bin20;
 ods output ParameterEstimates=temp.ph_bin20;
run;

* Truncated weights;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_norm_hom_T;
 ods output ParameterEstimates=temp.ph_norm_hom_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_t_hom1_T;
 ods output ParameterEstimates=temp.ph_t_hom1_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_t_hom5_T;
 ods output ParameterEstimates=temp.ph_t_hom5_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_gamma_T;
 ods output ParameterEstimates=temp.ph_gamma_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_bin10_T;
 ods output ParameterEstimates=temp.ph_bin10_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth survey_year urban_rural;
 model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban
_rural survey_year  / RL ;
 weight sw_bin20_T;
 ods output ParameterEstimates=temp.ph_bin20_T;
run;
```

```
proc export data= temp.ph_control outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - Control covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_norm_hom outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - Norm Hom covs.csv" dbms=csv replace
;
run;
proc export data= temp.ph_t_hom1 outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - T1 covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_t_hom5 outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - T5 covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_gamma outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - Gamma covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_bin1O outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - Bin1O covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_bin2O outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - Bin2O covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_norm_hom_T outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - T Norm Hom covs.csv" dbms=csv rep
lace;
run;
proc export data= temp.ph_t_hom1_T outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - T T1 covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_t_hom5_T outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - T T5 covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_gamma_T outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - T Gamma covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_bin1O_T outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - T Bin1O covs.csv" dbms=csv replace;
run;
proc export data= temp.ph_bin2O_T outfile="Z:\Output for Review\CP - Age Bins DRCPHM Output - T Bin2O covs.csv" dbms=csv replace;
run;
proc datasets library=temp;
 delete ph_control ph_norm_hom ph_t_hom1 ph_t_hom5 ph_gamma ph_bin1O ph_bin2O
   ph_norm_hom_T ph_t_hom1_T ph_t_hom5_T ph_gamma_T ph_bin1O_T ph_bin2O_T;
run;
quit;
run;

* MSMs;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight counter;
 ods output ParameterEstimates=temp.cmsm_control;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_norm_hom;
 ods output ParameterEstimates=temp.cmsm_norm_hom;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_t_hom1;
 ods output ParameterEstimates=temp.cmsm_t_hom1;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_t_hom5;
 ods output ParameterEstimates=temp.cmsm_t_hom5;
run;
```

```
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_gamma;
 ods output ParameterEstimates=temp.cmsm_gamma;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_bin10;
 ods output ParameterEstimates=temp.cmsm_bin10;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_bin20;
 ods output ParameterEstimates=temp.cmsm_bin20;
run;

* Truncated weights;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_norm_hom_T;
 ods output ParameterEstimates=temp.cmsm_norm_hom_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_t_hom1_T;
 ods output ParameterEstimates=temp.cmsm_t_hom1_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_t_hom5_T;
 ods output ParameterEstimates=temp.cmsm_t_hom5_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_gamma_T;
 ods output ParameterEstimates=temp.cmsm_gamma_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_bin10_T;
 ods output ParameterEstimates=temp.cmsm_bin10_T;
run;
proc phreg data=temp.dataset covs;
 strata age_buckets male race_eth;
 class  race_eth;
 model survival_time*cardiopulm_mort(0) = pm / RL ;
 weight sw_bin20_T;
 ods output ParameterEstimates=temp.cmsm_bin20_T;
```

```
run;
proc export data= temp.cmsm_control outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - Control covs.csv" dbms=csv replace
;
run;
proc export data= temp.cmsm_norm_hom outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - Norm Hom covs.csv" dbms=csv repla
ce;
run;
proc export data= temp.cmsm_t_hom1 outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - T1 covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_t_hom5 outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - T5 covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_gamma outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - Gamma covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_bin1O outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - Bin1O covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_bin2O outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - Bin2O covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_norm_hom_T outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - T Norm Hom covs.csv" dbms=csv r
eplace;
run;
proc export data= temp.cmsm_t_hom1_T outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - T T1 covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_t_hom5_T outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - T T5 covs.csv" dbms=csv replace;
run;
proc export data= temp.cmsm_gamma_T outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - T Gamma covs.csv" dbms=csv replace
;
run;
proc export data= temp.cmsm_bin1O_T outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - T Bin1O covs.csv" dbms=csv replace
;
run;
proc export data= temp.cmsm_bin2O_T outfile="Z:\Output for Review\CP - Age Bins PH MSM Output - T Bin2O covs.csv" dbms=csv replace
;
run;
proc datasets library=temp;
 delete cmsm_control cmsm_norm_hom cmsm_t_hom1 cmsm_t_hom5 cmsm_gamma cmsm_bin1O cmsm_bin2O
   cmsm_norm_hom_T cmsm_t_hom1_T cmsm_t_hom5_T cmsm_gamma_T cmsm_bin1O_T cmsm_bin2O_T;
run;
quit;
run;



*************************************************************************************************;
* Bootstrapping Approach;
*************************************************************************************************;

* The code below shows the bootstrapping approach for the normal homoskedastic weights - for other
 distributions, just change the code in the weight construction (using the code as written above)
 and all references to weights (such as changing sw_norm_hom to sw_gamma);

%let num_reps=1OO;
%let samp_fraction=1;
data boot_sub;
 set temp.dataset;
 keep pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_year
  age_buckets male race_eth mortstat cardiopulm_mort survival_time age;
run;
proc surveyselect data=boot_sub
 seed=123089
 out=dataset_boot
 method=urs
 samprate=&samp_fraction.
 reps=&num_reps.;
run;
```

```
ods select none;
data dataset_boot;
 set dataset_boot;
 counter=1;
run;
data boot_est_AC; *Create blank datasets to be filled with point estimates - for models with controls for covariates;
run;
data boot_est_CP;
run;
data boot_est_AC_T;
run;
data boot_est_CP_T;
run;
data boot_est_AC_m; *Create blank datasets to be filled with point estimates - for MSMs;
run;
data boot_est_CP_m;
run;
data boot_est_AC_T_m;
run;
data boot_est_CP_T_m;
run;
%macro boot_estimator_norm;
%do i=1 %to &num_reps.;
 data dataset_boot_sub;
  set dataset_boot;
  where replicate=&i.;
 run;

 * DENOMINATOR;
 proc genmod data=dataset_boot_sub;
  class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural survey_year ;
  model pm = age male race_eth smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural survey_
year  / maxiter=100;
  *variance of pm is sse/df = valuedf;
  ods output Modelfit=ss(where=(criterion="Deviance"));
  output out=x xbeta=xb;*predicted mean is xb;
 run;

 * Match dataset with valuedf (variance of pm);
 data ss;
  set ss;
  merg=1;
 run;
 data x;
  set x;
  merg=1;
 run;

 * Generate denominator density values;
 data dataset_boot_sub;
   merge x ss;
   by merg;
   sd_pm = sqrt(valuedf); *standard deviation of pm;
   pdf_den_denom = pdf("normal",pm,xb,sd_pm);*denominator density value;

   drop criterion df value valuedf xb sd_pm;
 run;

 * NUMERATOR;
 proc genmod data=dataset_boot_sub;
  model pm = ;
  *variance of pm is sse/df = valuedf;
  ods output Modelfit=ss(where=(criterion="Deviance"));
  output out=x xbeta=xb;*predicted mean is xb;
```

```
run;

* Match dataset with valuedf (variance of pm);
data ss;
 set ss;
 merg=1;
run;
data x;
 set x;
 merg=1;
run;

* Generate numerator density values;
data dataset_boot_sub;
  merge x ss;
  by merg;
  sd_pm = sqrt(valuedf); *standard deviation of pm;
  pdf_den_numer = pdf("normal",pm,xb,sd_pm); *numerator density value;
  drop criterion df value valuedf;
  * Creating weights;
  sw_norm_hom = pdf_den_numer / pdf_den_denom;
  drop pdf_den_numer pdf_den_denom xb sd_pm;
run;

* Generate truncated weights;
proc means data = dataset_boot_sub p1 p99;;
 var sw_norm_hom;
 output out = temp.weight_means p1=perc_1sw p99=perc_99sw;
run;

data temp.weight_means;
 set temp.weight_means;
 counter = 1;
run;

data dataset_boot_sub;
 merge dataset_boot_sub temp.weight_means;
 by counter;
 sw_norm_hom_T = sw_norm_hom;
 if sw_norm_hom < perc_1sw then sw_norm_hom_T = perc_1sw;
 if sw_norm_hom > perc_99sw then sw_norm_hom_T = perc_99sw;
run;

*****************************************************************;
*ANALYSIS;
* Standard weights;
*All-cause mortality;
proc phreg data=dataset_boot_sub;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural;
 model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
survey_year  / RL ;
 freq NumberHits;
 weight sw_norm_hom;
 hazardratio pm;
 ods output ParameterEstimates=est_rep_ac;
run;
data boot_est_AC;
 set boot_est_AC est_rep_ac;
run;

*Cardiopulmonary mortality;
proc phreg data=dataset_boot_sub;
 strata age_buckets male race_eth;
```

```
  class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural;
  model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urba
n_rural survey_year  / RL ;
  freq NumberHits;
  weight sw_norm_hom;
  hazardratio pm;
  ods output ParameterEstimates=est_rep_cp;
 run;
 data boot_est_CP;
  set boot_est_CP est_rep_cp;
 run;

 * Truncated weights;
 *All-cause mortality;
 proc phreg data=dataset_boot_sub;
  strata age_buckets male race_eth;
  class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural;
  model survival_time*mortstat(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urban_rural
 survey_year  / RL ;
  freq NumberHits;
  weight sw_norm_hom_T;
  hazardratio pm;
  ods output ParameterEstimates=est_rep_ac_T;
 run;
 data boot_est_AC_T;
  set boot_est_AC_T est_rep_ac_T;
 run;

 *Cardiopulmonary mortality;
 proc phreg data=dataset_boot_sub;
  strata age_buckets male race_eth;
  class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural;
  model survival_time*cardiopulm_mort(0) = pm smoking_status bmi_buckets income_buckets school_buckets marital_status nregion urba
n_rural survey_year  / RL ;
  freq NumberHits;
  weight sw_norm_hom_T;
  hazardratio pm;
  ods output ParameterEstimates=est_rep_cp_T;
 run;
 data boot_est_CP_T;
  set boot_est_CP_T est_rep_cp_T;
 run;

 ****************************************************************;
 * MSMs
 * Standard weights
 *All-cause mortality;
 proc phreg data=dataset_boot_sub;
  strata age_buckets male race_eth;
  class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural;
  model survival_time*mortstat(0) = pm   / RL ;
  freq NumberHits;
  weight sw_norm_hom;
  hazardratio pm;
  ods output ParameterEstimates=est_rep_ac_m;
 run;
 data boot_est_AC_m;
  set boot_est_AC_m est_rep_ac_m;
 run;

 *Cardiopulmonary mortality;
 proc phreg data=dataset_boot_sub;
  strata age_buckets male race_eth;
  class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural;
```

```
 model survival_time*cardiopulm_mort(0) = pm   / RL ;
 freq NumberHits;
 weight sw_norm_hom;
 hazardratio pm;
 ods output ParameterEstimates=est_rep_cp_m;
run;
data boot_est_CP_m;
 set boot_est_CP_m est_rep_cp_m;
run;

* Truncated weights
*All-cause mortality;
proc phreg data=dataset_boot_sub;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural;
 model survival_time*mortstat(0) = pm   / RL ;
 freq NumberHits;
 weight sw_norm_hom_T;
 hazardratio pm;
 ods output ParameterEstimates=est_rep_ac_T_m;
run;
data boot_est_AC_T_m;
 set boot_est_AC_T_m est_rep_ac_T_m;
run;

*Cardiopulmonary mortality;
proc phreg data=dataset_boot_sub;
 strata age_buckets male race_eth;
 class smoking_status bmi_buckets income_buckets school_buckets marital_status nregion race_eth urban_rural;
 model survival_time*cardiopulm_mort(0) = pm   / RL ;
 freq NumberHits;
 weight sw_norm_hom_T;
 hazardratio pm;
 ods output ParameterEstimates=est_rep_cp_T_m;
run;
data boot_est_CP_T_m;
 set boot_est_CP_T_m est_rep_cp_T_m;
run;

%end;
%mend;

* Estimate;
%boot_estimator_norm;

* Place estimates in datasets;
data temp.boot_est_AC;
 set boot_est_AC;
 if parameter ^= "pm" then delete;
run;
data temp.boot_est_CP;
 set boot_est_CP;
 if parameter ^= "pm" then delete;
run;
data temp.boot_est_AC_T;
 set boot_est_AC_T;
 if parameter ^= "pm" then delete;
run;
data temp.boot_est_CP_T;
 set boot_est_CP_T;
 if parameter ^= "pm" then delete;
run;
*MSMs;
data temp.boot_est_AC_m;
```

```
 set boot_est_AC_m;
 if parameter ^= "pm" then delete;
run;
data temp.boot_est_CP_m;
 set boot_est_CP_m;
 if parameter ^= "pm" then delete;
run;
data temp.boot_est_AC_T_m;
 set boot_est_AC_T_m;
 if parameter ^= "pm" then delete;
run;
data temp.boot_est_CP_T_m;
 set boot_est_CP_T_m;
 if parameter ^= "pm" then delete;
run;
proc export data= temp.boot_est_AC outfile="Z:\Output for Review\AC - Bootstrapped CPH Norm Hom.csv" dbms=csv replace;
run;
proc export data= temp.boot_est_CP outfile="Z:\Output for Review\CP - Bootstrapped CPH Norm Hom.csv" dbms=csv replace;
run;
proc export data= temp.boot_est_AC_T outfile="Z:\Output for Review\AC - Bootstrapped CPH Norm Hom_T.csv" dbms=csv replace;
run;
proc export data= temp.boot_est_CP_T outfile="Z:\Output for Review\CP - Bootstrapped CPH Norm Hom_T.csv" dbms=csv replace;
run;
*MSMs;
proc export data= temp.boot_est_AC_m outfile="Z:\Output for Review\AC - Bootstrapped CPH Norm Hom_m.csv" dbms=csv replace;
run;
proc export data= temp.boot_est_CP_m outfile="Z:\Output for Review\CP - Bootstrapped CPH Norm Hom_m.csv" dbms=csv replace;
run;
proc export data= temp.boot_est_AC_T_m outfile="Z:\Output for Review\AC - Bootstrapped CPH Norm Hom_T_m.csv" dbms=csv replace;
run;
proc export data= temp.boot_est_CP_T_m outfile="Z:\Output for Review\CP - Bootstrapped CPH Norm Hom_T_m.csv" dbms=csv replace;
run;
proc datasets library=temp;
 delete boot_est_AC_T boot_est_CP_T boot_est_AC_m boot_est_CP_m boot_est_AC_T_m boot_est_CP_T_m;
run;
quit;
run;
```