Measuring Health Inequalities: A Toolkit

Calculating Stratified Rates and Inequality Measures: Methodology and Code in SAS and R

Canadian Institute for Health Information
Institut canadien d'information sur la santé

Production of this document is made possible by financial contributions from Health Canada and provincial and territorial governments. The views expressed herein do not necessarily represent the views of Health Canada or any provincial or territorial government.

All rights reserved.

The contents of this publication may be reproduced unaltered, in whole or in part and by any means, solely for non-commercial purposes, provided that the Canadian Institute for Health Information is properly and fully acknowledged as the copyright owner. Any reproduction or use of this publication or its contents for any commercial purpose requires the prior written authorization of the Canadian Institute for Health Information. Reproduction or use that suggests endorsement by, or affiliation with, the Canadian Institute for Health Information is prohibited.

For permission or information, please contact CIHI:

Canadian Institute for Health Information
495 Richmond Road, Suite 600
Ottawa, Ontario K2A 4H6
Phone: 613-241-7860
Fax: 613-241-8120
cihi.ca
copyright@cihi.ca

ISBN 978-1-77479-110-3 (PDF)

© 2022 Canadian Institute for Health Information

How to cite this document:
Canadian Institute for Health Information. *Measuring Health Inequalities: A Toolkit — Calculating Stratified Rates and Inequality Measures: Methodology and Code in SAS and R*. Ottawa, ON: CIHI; 2022.

Cette publication est aussi disponible en français sous le titre *Mesurer les inégalités en santé : trousse d'outils — calcul des taux stratifiés et des mesures de l'inégalité : méthodologie et codes en SAS et en R*.
ISBN 978-1-77479-111-0 (PDF)

Table of contents

Terms of use for SAS and R codes

You may reproduce, modify, distribute and use the SAS and R codes included in appendices B, C, D and E of this publication (the **Codes**) at no charge for non-commercial purposes, including academic or government research or study purposes, related to building capacity for equity reporting in Canada so long as

- CIHI is properly and fully acknowledged as the copyright owner of the Codes;
- These terms of use are included with all whole or partial copies of the Codes, including any copies of modified versions of the Codes;
- You clearly identify any modifications made by you to the Codes;
- You do not use the Codes in a commercial manner, e.g., incorporating into a product for sale; and
- You do not state or suggest that CIHI uses, approves of, endorses or is affiliated with any person, entity, service, product, cause or opinion.

You further acknowledge and agree that

- Non-compliance with these terms of use may result in consequences that include but are not limited to the following: revocation of the rights granted to you under these terms of use and a requirement to destroy or return the Codes and all whole or partial copies.
- You will exercise caution or even refrain from publishing data generated using the Codes if the data identifies or could identify an individual or an individual health care facility.
- These terms of use constitute the entire agreement relating to the Codes.
- CIHI holds the rights in and title to the Codes. You do not obtain their ownership or any other right in them except as provided above.
- You will not take any action that may jeopardize CIHI's intellectual property rights in the Codes, or the application of these terms of use to the Codes.
- THE CODES ARE PROVIDED AS-IS, WITHOUT WARRANTIES OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING AS TO MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT WILL CIHI BE LIABLE FOR ANY DAMAGES OR LOSSES WHATSOEVER THAT MAY ARISE OUT OF OR IN CONNECTION WITH THE CODES.
- YOU WILL HOLD CIHI HARMLESS IN RESPECT OF ANY LOSSES OR DAMAGES ARISING FROM YOUR USE OF THE CODES.
- These terms are governed by, and to be construed in accordance with, the laws of the Province of Ontario and the applicable laws of Canada, without regard to principles of conflicts of laws.

You indicate your understanding and acceptance of these terms of use when you first use the Codes. IF YOU CANNOT ACCEPT THESE TERMS OF USE, YOU MAY NOT USE THE CODES.

Introduction

This guide provides an overview of the SAS and R code produced at the Canadian Institute for Health Information (CIHI) for calculating stratified crude and age-standardized health indicator rates (rates stratified by income quintiles, urban and rural/remote geographic location, etc.) and for calculating 4 summary measures of inequality: rate ratio, rate difference, potential rate reduction and population impact number. The methodology explains the approach used for both the rate and inequality calculations and applies to both the SAS and R code.

Methodology notes

Age-standardized rates

CIHI calculates age-standardized indicator rates by the direct method of standardization, using the 2011 Canadian population (from the 2011 Census) as the standard population. Standardization is based on 5-year age groupings.

The **age-standardized rate** is calculated as the sum of all age-specific weighted rates:

$$\text{Age-standardized rate} = \sum \text{Age-specific weighted rates}$$

The **age-specific weighted rates** are calculated for each age group using the following formula:

$$\text{Age-specific weighted rate} = \text{Rate}_{\text{crude}} \times \text{Weight} \times \text{Multiplier}$$

where $\text{Rate}_{\text{crude}}$ is calculated for each age group as the number of indicator cases divided by the population, Weight is the weight of each age group in the standard population, and Multiplier is the value for which you would like to present rates (e.g., per 100,000 population).

The **variance** of the age-specific weighted rate is based on a binomial distribution and is calculated for each age group using the following formula:

$$\text{Variance (Age-specific weighted rate)} = \text{Weight}^2 \times \text{Rate}_{\text{crude}} \times \frac{(\text{Multiplier} - (\text{Rate}_{\text{crude}}))}{\text{Age-specific population}}$$

The variance of the age-standardized rate is then calculated as the sum of the variances for all age-specific weighted rates:

$$\text{Variance (Age-standardized rate)} = \sum \text{Variance (Age-specific weighted rates)}$$

Simple inequality measures

Rate ratio (RR) is a measure of the relative inequality between subgroups that is calculated by dividing the rate of the comparison group by the rate of the reference group.

$$\text{Rate Ratio}_{\text{Comparison}} = \text{Rate}_{\text{Comparison}} \div \text{Rate}_{\text{Reference}}$$

Example: Measuring income-related inequalities, where quintile 1 (Q1) is the lowest income quintile (i.e., the comparison group) and quintile 5 (Q5) is the highest income quintile (i.e., the reference group). The rate in Q1 is 10 cases per 100,000 and the rate in Q5 is 5 cases per 100,000.

$\text{Rate}_{Q1} \div \text{Rate}_{Q5}$ = 10 per 100,000 ÷ 5 per 100,000 = 2

Interpretation: The rate of condition X is 2 times higher for Canadians in the lowest income quintile than for those in the highest income quintile.

The **variance** of $\text{Log}(\text{RR}_{\text{Comparison}})$ is calculated using the following formula:[1, i]

$$\text{Variance}\left(\text{Log}\left(\text{RR}_{\text{Comparison}}\right)\right) = \frac{\text{Variance}(\text{Rate}_{\text{Comparison}})}{{\text{Rate}_{\text{Comparison}}}^2} + \frac{\text{Variance}(\text{Rate}_{\text{Reference}})}{{\text{Rate}_{\text{Reference}}}^2}$$

The $\mathbf{RR_{Comparison}}$ **95% confidence interval** is given by $e^{\left(\text{Log}\left(\text{RR}_{\text{Comparison}}\right) \pm 1.96\sqrt{\text{Variance}\left(\text{Log}(\text{RR}_{\text{Comparison}})\right)}\right)}$.

Rate difference (RD) is a measure of the *absolute* inequality between subgroups that is calculated by subtracting the rate of the reference group from the rate of the comparison group.

Example: $\text{Rate}_{Q1} - \text{Rate}_{Q5}$ = 10 per 100,000 − 5 per 100,000 = 5 per 100,000

Interpretation: There are 5 more Canadians per 100,000 with condition X in the lowest-income quintile than in the highest-income quintile.

The **variance** of the rate difference is calculated using the following formula:

$$\text{Variance}\left(\text{RD}_{\text{Comparison}}\right) = \text{Variance}\left(\text{Rate}_{\text{Comparison}}\right) + \text{Variance}\left(\text{Rate}_{\text{Reference}}\right)$$

The $\mathbf{RD_{Comparison}}$ **95% confidence interval** is given by $\mathbf{RD_{Comparison} \pm 1.96\sqrt{\text{Variance}\left(RD_{Comparison}\right)}}$.

i. Formulas for variance and 95% confidence intervals are provided for administrative data and may need to be modified for use with other data sources (e.g., survey data will need to account for sampling strategy).

Complex inequality measures

Potential rate reduction and population impact number are 2 examples of a complex inequality measure. This type of measure can incorporate data from all population subgroups[ii] (e.g., inequality across all income quintiles), resulting in a single number indicating the level of inequality.

Potential rate reduction (PRR) is a *relative* measure of the potential reduction in a health indicator rate that would occur in the hypothetical scenario that each population subgroup experienced the same rate as the subgroup with the most desirable rate. It is also commonly known as the **population-attributable fraction** or **population-attributable risk**.[2] You can use the PRR in scenarios where lower indicator rates are desirable and where the reference group is the subgroup with the most desirable rate.

> *Example:* PRR = 45%
>
> *Interpretation:* In a given year, 45% of hospitalizations related to condition X could have been avoided if Canadians in all income quintiles had experienced the same rate of hospitalizations as those in the highest income quintile. (See Calculating and interpreting the PRR and PIN.)

The PRR and its **95% confidence interval** are calculated in the following manner for a stratifier with n categories (e.g., for a 5-category stratifier like income quintiles, n = 5). The PRR is commonly presented as a percentage, as follows:

$$\text{PRR} = \frac{\sum_{i=1}^{n} P_i \left(\frac{\text{Rate}_i}{\text{Rate}_{\text{Reference}}} - 1\right)}{1 + \sum_{i=1}^{n} P_i \left(\frac{\text{Rate}_i}{\text{Rate}_{\text{Reference}}} - 1\right)} \times 100\%$$

where P_i is the proportion of the population in the *i*th category.

ii. There are other ways to calculate potential rate reduction (also known as population-attributable fraction) for measuring health inequalities. For example, in the Public Health Agency of Canada's Health Inequalities Data Tool, population-attributable fractions are calculated for each comparison group rather than as a single number indicating inequality across all population subgroups.

The PRR's lower confidence interval (LCI) and upper confidence interval (UCI) are then given as follows:

$$\text{LCI}_{\text{PRR}} = 1 - \frac{1}{P_{\text{n}} + e^{\left(\log\left(\text{Rate ratio}_{1:(\text{n}-1),\,\text{n}}\right) - 1.96\sqrt{\text{Variance}\left(\text{Rate ratio}_{1:(\text{n}-1),\,\text{n}}\right)}\right)}}$$

$$\text{UCI}_{\text{PRR}} = 1 - \frac{1}{P_{\text{n}} + e^{\left(\log\left(\text{Rate ratio}_{1:(\text{n}-1),\,\text{n}}\right) + 1.96\sqrt{\text{Variance}\left(\text{Rate ratio}_{1:(\text{n}-1),\,\text{n}}\right)}\right)}}$$

where Rate $\text{ratio}_{1:(\text{n}-1),\,\text{n}}$

is the ratio of the sum of the population proportion (P_i) multiplied by the age-standardized rate (Rate_i) in the first (n − 1) categories relative to the rate in the reference category, calculated as

$$\frac{\sum_{i=1}^{(\text{n}-1)} P_i \text{Rate}_i}{\text{Rate}_{\text{Reference}}}$$

with variance

$$\frac{\sum_{i=1}^{(\text{n}-1)} P_i^2 \times \text{Variance}\left(\text{Rate}_i\right)}{\left(\sum_{i=1}^{(\text{n}-1)} P_i \text{Rate}_i\right)^2} + \frac{\text{Variance}\left(\text{Rate}_{\text{Reference}}\right)}{\text{Rate}_{\text{Reference}}^{\;2}}$$

Population impact number (PIN) is an *absolute* measure of the potential reduction in the number of cases for a health indicator that would occur in the hypothetical scenario that each population subgroup experienced the same rate as the subgroup with the most desirable rate. It captures the gradient of inequality across multiple categories, such as income quintiles.

Example: PIN = 18,700

Interpretation: In a given year, approximately 18,700 hospitalizations related to condition X could have been avoided if Canadians in all income levels had experienced the same rate of hospitalizations as those in the highest income level. (See Calculating and interpreting the PRR and PIN.)

The PIN is related to the PRR in the following manner:

$$\text{Population impact number} = \text{Overall indicator standardized rate} \times N_{\text{total population}} \times \text{PRR}$$

Potential rate improvement (PRI) is a measure of *relative* inequality (analogous to the PRR) used in scenarios where higher indicator rates are desirable. It is also commonly known as the prevented fraction.

The PRI is related to the PRR in the following manner:

$$PRI = 1 - \frac{1}{1 - \text{PRR}}$$

where PRR refers to the potential rate reduction expressed as a decimal.

Drawing on the variance calculation for the PRR, the variance of the PRI is given by the following equation:

$$Variance(PRI) = variance(1 - \frac{1}{1 - \text{PRR}})$$

$$Variance(PRI) = variance(\log(\frac{Rate_{1:(n-1)}}{Rate_{Reference}})) = \frac{variance(Rate_{Reference})}{(Rate_{Reference})^2} + \frac{\sum_{i=1}^{(n-1)} P_i^2\, variance(Rate_i)}{(Rate_{1:(n-1)})^2}$$

The lower and upper 95% confidence intervals are then given as follows:

$$LCI_{PRI} = Exp(log(PRI) - 1.96\sqrt{variance}(PRI))$$

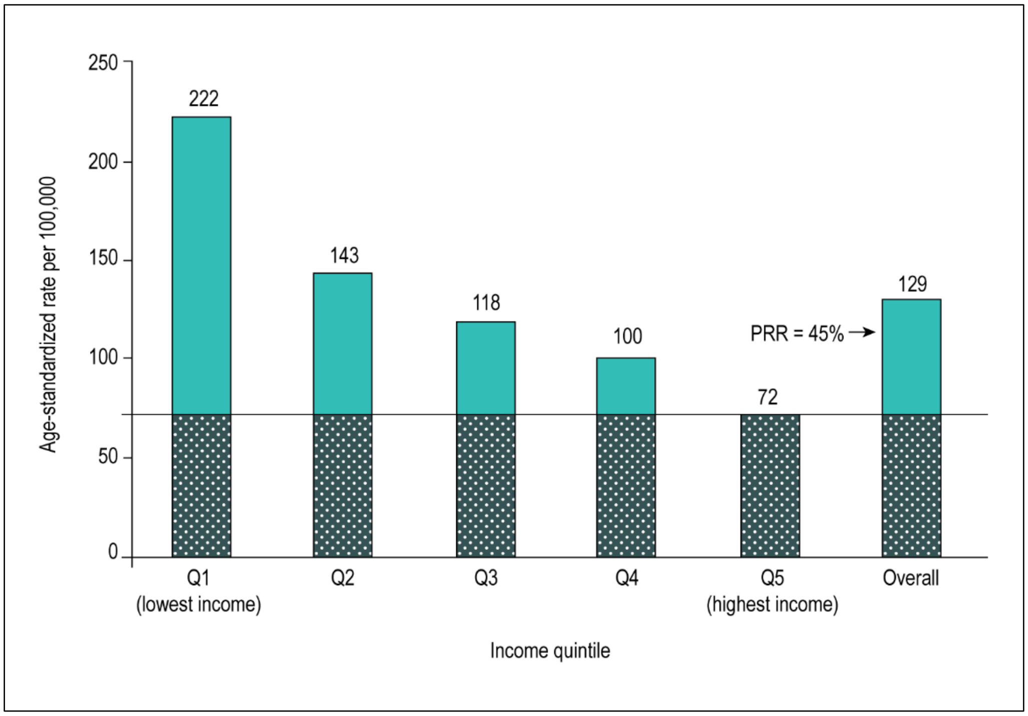
$$UCI_{PRI} = Exp(log(PRI) + 1.96\sqrt{variance}(PRI))$$

Calculating and interpreting the PRR and PIN

Let's look at an example using the indicator Chronic Obstructive Pulmonary Disease (COPD) Hospitalization for Canadians Younger Than Age 75 to understand how to calculate and interpret the PRR and PIN.

When stratified by income quintile, the age-standardized hospitalization rates for this indicator range from a low of 72 per 100,000 in the highest income level (Q5) to a high of 222 per 100,000 in the lowest income level (Q1).

Figure COPD hospitalization rates for Canadians younger than 75, by income, 2012



Calculating the PRR

When we calculate a PRR, we are interested in estimating how much lower the overall rate would be if all income quintiles had the rate of the group with the most desirable rate (i.e., the highest income quintile).

In the figure, the horizontal line represents the indicator rate for the highest income level (Q5).

To calculate a PRR, you need the following pieces of information:

- The age-standardized rates for each income quintile (Rate_i)
- The proportion of the population in each income quintile (P_i) (see the box below)

Insert the values into the PRR formula as follows:

$$\text{PRR} = \frac{\sum_{i=1}^{5} P_i \left(\frac{\text{Rate}_i}{\text{Rate}_5} - 1\right)}{1 + \sum_{i=1}^{5} P_i \left(\frac{\text{Rate}_i}{\text{Rate}_5} - 1\right)} \times 100\%$$

$$\text{PRR} = \frac{\left[0.20\left(\frac{222}{72}-1\right) + 0.20\left(\frac{143}{72}-1\right) + 0.19\left(\frac{118}{72}-1\right) + 0.21\left(\frac{100}{72}-1\right) + 0.20\left(\frac{72}{72}-1\right)\right]}{\mathbf{1} + \left[0.20\left(\frac{222}{72}-1\right) + 0.20\left(\frac{143}{72}-1\right) + 0.19\left(\frac{118}{72}-1\right) + 0.21\left(\frac{100}{72}-1\right) + 0.20\left(\frac{72}{72}-1\right)\right]} \times \mathbf{100}\%$$

PRR = 45%

Interpreting the PRR

In 2012, about 45% of COPD hospitalizations among Canadians younger than 75 could have been avoided if the rate of COPD hospitalizations in all income quintiles had been the same low rate as the rate for the highest income quintile.

Box: Understanding the proportion of the population in each income quintile (Pi)

By definition, the proportion of the population in each quintile is equal to 20%. However, for this COPD hospitalization indicator, the proportion of the population in each income quintile is not exactly 20%.

This is because the proportion of the population in each income quintile for this indicator is derived from census population counts for Canadians younger than 75 (i.e., the denominator count for each quintile divided by the total population age 0 to 74).

These census population counts have been assigned to income quintiles that were constructed using the full Canadian population (i.e., all ages). As such, they reflect the income distribution of all Canadians, not the specific income distribution of Canadians younger than 75.

In the PRR calculation above, the proportion of the population in each income quintile is as follows: P1 = 0.20, P2 = 0.20, P3 = 0.19, P4 = 0.21 and P5 = 0.20.

Calculating the PIN

When we calculate a PIN, we are converting the PRR to the approximate number of cases that could be avoided in the hypothetical scenario where all population subgroups experience the same rate as the subgroup with the most desirable rate.

To calculate a PIN, you need the following pieces of information:

- The overall indicator rate
- The PRR
- The total number of people in the population (for this analysis, the population of Canadians age 0 to 74 in 2012 is 32,079,232)

Insert the values into the PIN formula as follows:

$$PIN = Overall\ indicator\ standardized\ rate \times N_{total\ population} \times PRR$$

= 129 ÷ 100,000 × 32,079,232 × 45 ÷ 100

PIN = 18,663

Interpreting the PIN

In 2012, there could have been approximately 18,700 fewer hospitalizations for COPD among Canadians younger than 75 if the rate of COPD hospitalizations in all income quintiles had been the same low rate as the rate in the highest income quintile.

Note: While the calculated value is 18,663, at CIHI we report the PIN as an approximate number (in this case, rounded to the nearest 100).

Considerations for reporting summary measures

CIHI uses 95% confidence intervals to determine the statistical significance of health inequality summary measures, as shown in the following table (note that other approaches are possible).

Inequality measure	Considered to be statistically significant if 95% confidence interval (CI)
Rate ratio (RR)	Does not include 1
Rate difference (RD)	Does not include 0
Potential rate reduction (PRR)	Does not include 0
Population impact number (PIN)	Does not include 0
Potential rate improvement (PRI)	Does not include 0

- CIHI's best practices for reporting summary measures of inequality include
  - Reporting both the PRR and the PIN as 0 if the PRR is negative and statistically different from 0 (i.e., upper and lower confidence limits are negative);
  - Suppressing and giving a value of 0 to PRI values that are negative and statistically significant; and
  - Reporting the PIN as an approximate number (e.g., rounded to the nearest 100).

Appendices

Appendix A: Code overview

Calculate_Stratified_Rates

This code calculates crude and age-standardized rates stratified by the equity stratifier of your choice (income quintile, urban and rural/remote, etc.). Stratified rates are also calculated by sex (male, female and both sexes combined) and reporting level (national and provincial/territorial). This code may be modified to calculate stratified rates for different jurisdictional levels.

To use the code,

- Create the following 3 input data sets using the format described in the Excel file *Measuring Health Inequalities: A Toolkit — Input File Formats for SAS Macros and R Functions*.
    1. Indicator cases
    2. Population estimates
    3. Standard population estimates

The Excel file contains fake sample data for these 3 input data sets, which can be used to run example analyses.

- Define all input parameters (e.g., age range for the indicator).
- Run Calculate_Stratified_Rates code (see Appendix B for SAS macro and Appendix D for R function).
- Check that your output is complete with no errors or warnings.

Note that variable types (numeric or character) should be as specified in the Excel file. Additional “Format” requirements presented in the Excel file are required for the SAS macros only.

Calculate_Inequality_Measures

This code calculates inequality measures — including rate ratio, rate difference, potential rate reduction and population impact number — for health indicator rates stratified by an equity stratifier (e.g., income quintile). Results are generated by sex (male, female and both sexes combined) and reporting level (national and provincial/territorial).

To use the code,

- Create an input data set that is formatted as described in the Excel file *Measuring Health Inequalities: A Toolkit — Input File Formats for SAS Macros and R Functions*. Note that the output data from successfully running Calculate_Stratified_Rates will be correctly formatted.
- Run the Calculate_Inequality_Measures code (see Appendix C for SAS macro and Appendix E for R function).
- Check that your output is complete with no errors or warnings.

Appendix B: SAS macro — %Calculate_Stratified_Rates

Output file

&indicator._&YR._&AGEGPL._&AGEGPU._&equity_stratifier._rates — Crude and age-standardized rates for the chosen equity stratifier and reporting level.

Parameters

You must define all parameters listed below.

- *indicator* — The name of the health indicator you are measuring. This can be any name you choose.
- *yr* — The year. Indicator cases and population estimates will be extracted only from the corresponding year. Only 1 year can be specified at a time.
- *reporting_level* — The geographic reporting level. The macro calculates rates nationally and for the provinces and territories. Use the same variable name for reporting_level in your infile and popfile data sets.
- *equity_stratifier* — The variable for which you would like to calculate stratified rates (e.g., income quintile). This can be an ordered stratifier, such as income quintile, or an unordered stratifier, such as urban and rural/remote geographic location. You should use the same variable name for equity_stratifier in all your input data sets.
- *inlib* — The name of the SAS library where you have stored your input data set.
- *infile* — The name of the input data set of indicator cases.
- *stdpop* — The name of the standard population estimates data set.
- *popfile* — The name of the population estimates data set categorized by your equity stratifier (e.g., population estimates by income quintile).
- *popvar* — The population counts from popfile.
- *age_var* — The name of the age variable used consistently in the infile, stdpop and popfile data sets.
- *agegpl* — The lower age limit of the health indicator; must be in format 01, 02 . . . 14, 15 as specified in the age variable (e.g., if the lower age limit for your health indicator is 0 years, this value would be 01).
- *agegpu* — The upper age limit of the health indicator; must be in format 01, 02 . . .14, 15 as specified in the age variable (e.g., if the upper age limit for your health indicator is 74 years, this value would be 15).

- *multiplier* — The value for which you would like to present rates (e.g., multiplier = 100,000 for rates expressed per 100,000).
- *roundunit* — The rounding unit (e.g., to round to 1 decimal place, specify ROUNDUNIT = .01).
- *outlib* — The name of the SAS library where you want your results output.
- *outfile* — The name of the output data set.

Macro invocation

%MACRO Calculate_Stratified_Rates (indicator=, yr=, reporting_level=, equity_stratifier=, inlib=, infile=, stdpop=, popfile=, popvar=, age_var=, agegpl=, agegpu=, multiplier=, roundunit=, outlib=, outfile=);

The example invocation below represents how the macro can be called if using the fake sample data provided in the Excel file. Example invocation assumes all files are stored in the *work* library.

```
%Calculate_Stratified_Rates (
        indicator=fake,
        yr=2020,
        reporting_level=PT_code,
        equity_stratifier=income_quintile,
        inlib=work,
        infile=Example_infile_data,
        stdpop=work.Example_stdpop_file,
        popfile=work.Example_popfile_data,
        popvar=PT_pop,
        age_var=AGE_GROUP_CODE,
        agegpl=01,
        agegpu=19,
        multiplier=100000,
        roundunit=.01,
        outlib=work,
        outfile=fakedata_output1);
```

Macro steps

1. Prepares indicator cases by creating aggregated counts for reporting level, age group, sex and equity stratifier.
2. Prepares the standard population file used for age standardization.
3. Prepares the population denominators for the age range of interest.
4. Links the 3 data sets listed above at the provincial/territorial level by age group, sex and equity stratifier.
5. Calculates stratified crude rates and uses the following steps to calculate stratified age-standardized rates:
   a) Calculates age-specific rates for each age group.
   b) Multiplies the age-specific rates of the population under study by the number of persons in each age group of the standard population to get the age-specific weighted rate for each age group.
   c) For each equity stratifier category, sums all age-specific weighted rates to get the age-standardized rate.
   d) Calculates variance and uses this to calculate confidence intervals.
6. Calculates overall crude and age-standardized rates. Note that overall rates refer to all categories combined within the equity stratifier (e.g., for the income stratifier, this refers to the overall rate for quintiles 1 through 5). For this reason, any cases that are not assigned to an equity stratifier category (e.g., due to missing postal code) will be excluded from the overall rate.
7. Outputs the data table combining overall rates and rates by equity stratifier.

Macro code

```
%MACRO Calculate_Stratified_Rates(indicator= , yr= , reporting_level= , equity_stratifier= ,
inlib=, infile=, stdpop=, popfile=, popvar=, age_var=, agegpl=, agegpu=, multiplier=, roundunit=,
outlib=, outfile=);

/****************************************************************/
/* STEP 1: Prepare data set of indicator cases aggregated by reporting level, sex, age group
/* and equity stratifier */
/****************************************************************/
/****************************************************************/
/* 1A. Create count variable for the indicator data set and output required year and age groups
/****************************************************************/
data &indicator.&yr._&AGEGPL._&AGEGPU.;
	set &inlib..&infile.;
	* Create a count variable for the indicator;
	&indicator.=1;
	* Output required year and age groups only;
	if year eq &yr. and ("&AGEGPL."<=&AGE_VAR.<="&AGEGPU.") then output;
run;

/****************************************************************/
/* 1B. Create aggregated indicator data sets - national and by province/territory */
/* Prepare aggregated counts from indicator data set (infile) by age group, sex
/* and equity stratifier.
/* Do this for Canada and by reporting level.
/****************************************************************/
* Sum indicator cases by sex, age group and equity stratifier for all of Canada;
proc sql;
	create table &indicator._canada as
	select sex, &AGE_VAR., &equity_stratifier., SUM(&indicator.) as cases
	from &indicator.&yr._&AGEGPL._&AGEGPU.
	group by sex, &AGE_VAR., &equity_stratifier.
	OUTER UNION CORR
	select "3" AS sex, &AGE_VAR., &equity_stratifier., SUM(&indicator.) as cases
	from &indicator.&yr._&AGEGPL._&AGEGPU.
	group by &AGE_VAR., &equity_stratifier.
	ORDER BY SEX, &AGE_VAR., &equity_stratifier.;
QUIT;

/* Define Canada as '99' */
data &indicator._canada;
	set &indicator._canada;
```

```
		&reporting_level.="99";
	run;

	* Sum indicator cases by sex, age group, equity stratifier and reporting level;
	proc sql;
		create table &indicator._&reporting_level. as
		select &reporting_level., sex, &AGE_VAR., &equity_stratifier., SUM(&indicator.)
		as cases
		from &indicator.&yr._&AGEGPL._&AGEGPU.
		group by &reporting_level., sex, &AGE_VAR., &equity_stratifier.
		OUTER UNION CORR
		select &reporting_level., "3" AS sex, &AGE_VAR., &equity_stratifier., SUM(&indicator.)
		as cases
		from &indicator.&yr._&AGEGPL._&AGEGPU.
		group by &reporting_level., &AGE_VAR., &equity_stratifier.
		ORDER BY &reporting_level., SEX, &AGE_VAR., &equity_stratifier.;
	QUIT;

	data &indicator._&reporting_level.;
		set &indicator._&reporting_level. &indicator._canada;
	run;

	/*****************************************************************/
	/* STEP 2. Prepare standard population file */
	/*****************************************************************/
	/*****************************************************************/
	/* 2A. Sum the standard population by indicator age groups within age range of the indicator
	/* and keep the total population
	/* Multiply each of the age-specific rates by the proportion of the 2011 population belonging
	/* to the particular age group (called the standard population weight)
	/*****************************************************************/
	proc sql;
		create table stdpop_agegroup as
		select &AGE_VAR., standard_pop_cnt as stdpop, sum(standard_pop_cnt) as
		agegroup_stdpop
		from &stdpop.(where=("&AGEGPL."<=&AGE_VAR.<="&AGEGPU."));
	quit;

	proc sort data=stdpop_agegroup; by &AGE_VAR.; run;

	proc sql;
		create table stdpop_population as
		select &AGE_VAR., sum(standard_pop_cnt) as total_stdpop
		from &stdpop.(where=("&AGEGPL."<=&AGE_VAR.<="&AGEGPU."));
```

```
quit;

proc sort data=stdpop_population; by &AGE_VAR.; run;

data stdpop_new;
	merge stdpop_agegroup (in=a) stdpop_population(in=b);
	by &AGE_VAR.;
	if a=1 then do;
		weight= stdpop/agegroup_stdpop;
		output;
	end;
	drop total_stdpop;
run;

/***************************************************************/
/* STEP 3. Prepare population data */
/***************************************************************/
/***************************************************************/
/* 3A. Prepare population denominators for the specific age range of indicator */
/***************************************************************/
data new_popfile;
	set &popfile.;
	if population_year eq &yr. and ("&AGEGPL."<=&AGE_VAR.<="&AGEGPU.") then output;
run;

/***************************************************************/
/* STEP 4. Join data sets from steps 1 to 3 to create datasets for calculating rates by equity
/* stratifier in Step 5 and overall rates in Step 6.
/***************************************************************/
/***************************************************************/
/* 4A. Create dataset containing population denominator and standard population datasets
/***************************************************************/
proc sql;
	create table new_popfile2 as
	select a.*, b.stdpop, b.weight
	from new_popfile as a
	right join stdpop_new as b
	on a.&AGE_VAR.=b.&AGE_VAR.
	order by &reporting_level., a.sex, b.&AGE_VAR., a.&equity_stratifier.;
quit;



/***************************************************************/
```

```
/* 4B. Join in dataset of aggregated indicator cases and roll up by equity stratifier,
/* reporting_level, sex and age.
/* This will be used for calculating rates by equity stratifier in Step 5.
/****************************************************************/
proc sql;
        create table count_pop_&equity_stratifier. as
        select a.cases as count, b.*
        from &indicator._&reporting_level. as a
        right join new_popfile2 as b
        on a.&AGE_VAR.=b.&AGE_VAR. and a.sex=b.sex and
        a.&equity_stratifier.= b.&equity_stratifier. and a.&reporting_level.=b.&reporting_level.
        order by &reporting_level., &equity_stratifier., sex, &AGE_VAR.;
quit;

/****************************************************************/
/* 4C. Create new population file rolling up all &equity_stratifer categories
/****************************************************************/
proc sql;
        create table new_popfile3 as
        select population_year, &reporting_level., sex, &AGE_VAR., sum(&popvar.) as
        &popvar., stdpop, weight
        from new_popfile2
        group by population_year, &reporting_level., sex, &AGE_VAR., stdpop, weight;
quit;

/****************************************************************/
/* 4D. Aggregate indicator cases by reporting level, sex and age only
/****************************************************************/
proc sql;
        create table &indicator._&reporting_level._all
        as select &reporting_level., sex, &AGE_VAR., sum(cases) as cases
        from &indicator._&reporting_level.
        group by &reporting_level., sex, &AGE_VAR.;
quit;

 /****************************************************************/
/* 4E. Join datasets from step 4C and 4D.
                        This will be used for calculating overall rates in Step 6.
/****************************************************************/
proc sql;
        create table count_pop as
        select a.cases as count, b.*
        from &indicator._&reporting_level._all as a
        right join new_popfile3 as b
```

```
        on a.&AGE_VAR.=b.&AGE_VAR. and a.sex=b.sex and
        a.&reporting_level.=b.&reporting_level.
        order by &reporting_level., sex, &AGE_VAR.;
quit;

/*****************************************************************/
/* STEP 5. Calculate crude and age-standardized rates stratified by equity stratifier
/*****************************************************************/
/*****************************************************************/
/* 5A. Calculate crude, expected, variance and standardized numerator count
/* Variance calculation is based on a binomial distribution
/* Standardized numerator count (std_num) is required for Calculate_Inequality_Measures */
/*****************************************************************/
%if &popvar. ne 0 %then %do;
        data &indicator._rates_&YR._&AGEGPL._&AGEGPU.;
                set count_pop_&equity_stratifier.;
                if count=. then count=0;
                crude=(count/&popvar.)*&MULTIPLIER.;
                expect=crude*weight;
                variance=((weight*weight)*crude*(&MULTIPLIER.-crude))/&popvar.;
                std_num=count/&popvar.*stdpop;
        run;
%end;
%if &popvar. = 0 %then %do;
        data &indicator._rates_&YR._&AGEGPL._&AGEGPU.;
                set count_pop_&equity_stratifier.;
                if count=. then count=0;
                crude=0;
                expect=crude*weight;
                variance=((weight*weight)*crude*(&MULTIPLIER.-crude))/&popvar.;
                std_num=count/&popvar.*stdpop;
                run;
%end;

/*****************************************************************/
/* 5B. Calculate age-standardized rates (ASR) by equity stratifier categories
/* Sum counts, total population, crude rate, standard numerator, expected and variance
/* by reporting level, sex and equity stratifier
/*****************************************************************/
proc sql;
        create table &indicator._ASR_&YR._&AGEGPL._&AGEGPU._&equity_stratifier. as
        select &reporting_level., sex, &equity_stratifier.,
        sum(count) as count, sum(&popvar.) as totalpop,
        sum(count)/sum(&popvar.)*&MULTIPLIER. as crude, sum(std_num) as std_numer,
```

```
        sum(expect) as sum_exp, sum(variance) as variance
        from &indicator._rates_&YR._&AGEGPL._&AGEGPU.
        group by &reporting_level., sex, &equity_stratifier.;
quit;

/****************************************************************/
/* 5C. Calculate confidence intervals for age-standardized rates by equity stratifier
/****************************************************************/
data &indicator._ASR_&YR._&AGEGPL._&AGEGPU._&equity_stratifier.;
        set &indicator._ASR_&YR._&AGEGPL._&AGEGPU._&equity_stratifier.;
        lci=sum_exp-1.96*sqrt(variance);
        uci=sum_exp+1.96*sqrt(variance);
        if lci<0 then do;
                p=sum_exp/&MULTIPLIER.;
                if p=0 then p=1/(&MULTIPLIER.*&MULTIPLIER.);
                logitp=log(p/(1-p));
                var=1/(p*p*(1-p)*(1-p))*(variance/(&MULTIPLIER.*&MULTIPLIER.));
                A=logitp-1.96*sqrt(var);
                B=logitp+1.96*sqrt(var);
                lci=&MULTIPLIER./(1+exp(-A));
                uci=&MULTIPLIER./(1+exp(-B));
                drop p logitp var A B;
        end;
        %if &ROUNDUNIT. = 1 %then format crude rstd rlci ruci 8.;
        %else   format asr asr_lci asr_uci 8&ROUNDUNIT.;;
                crude_rate= round(crude,&ROUNDUNIT.);
                asr = round(sum_exp,&ROUNDUNIT.);
                asr_lci = round(lci,&ROUNDUNIT.);
                asr_uci = round(uci,&ROUNDUNIT.);
run;

/****************************************************************/
/* STEP 6. Calculate overall rates */
/****************************************************************/
/****************************************************************/
/* 6A. Calculate crude, expected, variance and standardized numerator count
/* Variance calculation is based on a binomial distribution
/*        Standardized numerator count (std_num) required in Calculate_Inequality_Measures*/
/****************************************************************/
%if &popvar. ne 0 %then %do;
        data &indicator._rates_&YR._&AGEGPL._&AGEGPU._ALL;
                set count_pop;
                if count=. then count=0;
                crude=(count/&popvar.)*&MULTIPLIER.;
```

```
        expect=crude*weight;
        variance=((weight*weight)*crude*(&MULTIPLIER.-crude))/&popvar.;
        std_num=count/&popvar.*stdpop;
    run;
%end;
%if &popvar. = 0 %then %do;
    data &indicator._rates_&YR._&AGEGPL._&AGEGPU._ALL;
        set count_pop;
        if count=. then count=0;
        crude=0;
        expect=crude*weight;
        variance=((weight*weight)*crude*(&MULTIPLIER.-crude))/&popvar.;
        std_num=count/&popvar.*stdpop;
    run;
%end;

/**************************************************************/
/* 6B. Calculate overall age-standardized rates
/* Sum counts, total population, crude rate, standard numerator, expected and
/* variance by reporting level and sex
/**************************************************************/
proc sql;
    create table &indicator._ASR_&YR._&AGEGPL._&AGEGPU. as
    select &reporting_level., sex,
    sum(count) as count, sum(&popvar.) as totalpop,
    sum(count)/sum(&popvar.)*&MULTIPLIER. as crude, sum(std_num) as std_numer,
    sum(expect) as sum_exp, sum(variance) as variance
    from &indicator._rates_&YR._&AGEGPL._&AGEGPU._ALL
    group by &reporting_level., sex;
quit;

/**************************************************************/
/* 6C. Calculate confidence intervals for overall age-standardized rates
/**************************************************************/
data &indicator._ASR_&YR._&AGEGPL._&AGEGPU.;
    set &indicator._ASR_&YR._&AGEGPL._&AGEGPU.;
    lci=sum_exp-1.96*sqrt(variance);
    uci=sum_exp+1.96*sqrt(variance);
    if lci<0 then do;
        p=sum_exp/&MULTIPLIER.;
        if p=0 then p=1/(&MULTIPLIER.*&MULTIPLIER.);
        logitp=log(p/(1-p));
        var=1/(p*p*(1-p)*(1-p))*(variance/(&MULTIPLIER.*&MULTIPLIER.));
        A=logitp-1.96*sqrt(var);
```

```
                B=logitp+1.96*sqrt(var);
                lci=&MULTIPLIER./(1+exp(-A));
                uci=&MULTIPLIER./(1+exp(-B));
                drop p logitp var A B;
        end;

        %if &ROUNDUNIT. = 1 %then
                format crude rstd rlci ruci 8.;
        %else
        format asr asr_lci asr_uci 8&ROUNDUNIT.;;
        crude_rate= round(crude,&ROUNDUNIT.);
        asr = round(sum_exp,&ROUNDUNIT.);
        asr_lci = round(lci,&ROUNDUNIT.);
        asr_uci = round(uci,&ROUNDUNIT.);
run;

/****************************************************************/
/* STEP 7. Output data set
/****************************************************************/
* ASR by equity stratifier;
data &indicator._ASR_&YR._&AGEGPL._&AGEGPU._&equity_stratifier.;
        retain year &reporting_level. sex &equity_stratifier count totalpop crude_rate asr
        asr_lci asr_uci variance  std_numer;
        set &indicator._ASR_&YR._&AGEGPL._&AGEGPU._&equity_stratifier.;
        year=&yr.;
        keep year &reporting_level. sex &equity_stratifier count totalpop crude_rate asr
        asr_lci asr_uci variance std_numer;
run;

* ASR overall;
data &indicator._ASR_&YR._&AGEGPL._&AGEGPU.;
        retain year &reporting_level. sex &equity_stratifier count totalpop crude_rate asr
        asr_lci asr_uci variance std_numer;
        set &indicator._ASR_&YR._&AGEGPL._&AGEGPU.;
        year=&yr.;
        &equity_stratifier ="0";
        keep year &reporting_level. sex &equity_stratifier count totalpop crude_rate asr
        asr_lci asr_uci variance std_numer;
run;

* Combine ASR overall and by equity stratifier;
data &indicator._&YR._&AGEGPL._&AGEGPU._&equity_stratifier._rates;
        set &indicator._ASR_&YR._&AGEGPL._&AGEGPU._&equity_stratifier.
        &indicator._ASR_&YR._&AGEGPL._&AGEGPU.;
```

```
        label year = 'Year'
                &reporting_level.= 'Reporting level'
                &equity_stratifier. = "&equity_stratifier. "
                sex= 'Sex'
                count = 'Number of indicator cases'
                totalpop = "Total population"
                crude_rate = "Crude rate per &MULTIPLIER.  population"
                variance = "Variance"
                asr = "Age-standardized rate per &MULTIPLIER. population"
                asr_lci  = "Lower confidence limit of the age-standardized rate"
                asr_uci = "Upper confidence limit of the age-standardized rate"
                std_numer ="Standard numerator";
                if  &equity_stratifier. eq . then delete;
run;

proc sort data=&indicator._&YR._&AGEGPL._&AGEGPU._&equity_stratifier._rates;
        by &reporting_level. sex &equity_stratifier.;
run;

/****************************************************************/
/* Note:        This data set is formatted to run in the Calculate_Inequality_Measures macro.
/****************************************************************/
data &outlib..&outfile.;
        set &indicator._&YR._&AGEGPL._&AGEGPU._&equity_stratifier._rates;
run;

%MEND Calculate_Stratified_Rates;
```

Appendix C: SAS macro — %Calculate_Inequality_Measures

Output files

3 output files will be created after running this macro program:

- rd_rr_&indicator.&yr_by_&equity_stratifier. — RR and RD results based on equity stratifier and reporting level
- prr_pin_&indicator.&yr_by_&equity_stratifier. — PRR and PIN results based on equity stratifier and reporting level
- &outlib..&outfile. — Final RR, RD, PRR and PIN results based on equity stratifier and reporting level

Parameters

You must define all parameters listed below unless otherwise specified.

- *data* — The name of the data set being input into the macro.
- *indicator* — The name of the health indicator you are measuring.
- *yr* — The year.
- *equity_stratifier* — The variable for which you would like to calculate inequality measures (e.g., income quintile).
- *ordered_data* — Specifies whether your equity stratifier represents ordered data (e.g., 5 income quintiles, where quintile 1 is the lowest income group and quintile 5 is the highest) or unordered data (e.g., language). The macro accepts 2 possible values: "yes" for ordered data or "no" for unordered data.
- *ordered_ref_group* — For ordered equity stratifiers, this parameter specifies whether the reference group is the highest or lowest group. The macro accepts 2 possible values: "highest" if the reference group is the highest group or "lowest" if the reference group is the lowest group. For unordered equity stratifiers, leave this parameter blank.
- *unordered_ref_group* — For unordered equity stratifiers, you must specify the reference group; this is a character value representing the group that all other groups are compared against. For ordered equity stratifiers, leave this parameter blank.
- *reporting_level* — The geographic reporting level.
- *sex* — The sex variable for the indicator; must include even if reporting for both sexes (use a dummy in its place).

- *multiplier* — The value for which you would like to present rates (e.g., multiplier = 100,000 for rates expressed per 100,000).
- *outlib* — The name of the SAS library where you want your results output.
- *outfile* — The name of the output data set.

Macro invocation

%MACRO Calculate_Inequality_Measures(data=, indicator=, yr=, equity_stratifier=, ordered_data=, ordered_ref_group=, unordered_ref_group=, reporting_level=, sex=, outlib=, outfile=, multiplier=);

The example invocation below represents how the macro can be called if using the output created from running the fake sample data provided in the Excel file through %Calculate_Stratified_Rates. Example invocation assumes all files are stored in the *work* library.

```
%Calculate_Inequality_Measures(
        data=work.fakedata_output1,
        indicator=fake,
        yr=2020,
        equity_stratifier=income_quintile,
        ordered_data=yes,
        ordered_ref_group=highest,
        unordered_ref_group=,
        reporting_level=PT_code,
        sex=sex,
        outlib=work,
        outfile=fakedata_output2,
        multiplier=100000);
```

Macro steps

1. Creates a data set containing the reference group for each combination of reporting_level and sex.
2. Calculates RR and RD.
3. Calculates PRR and PIN.
4. Outputs data tables with all summary measures of inequality.

Macro code

```
%MACRO Calculate_Inequality_Measures(data=, indicator=, yr=, equity_stratifier=,
        ordered_data=, ordered_ref_group=, unordered_ref_group=,
        reporting_level=, sex=, outlib=, outfile=, multiplier=);

/*****************************************************************/
/* STEP 1: For both ordered and unordered data, create a dataset containing
/* the reference group for each combination of reporting level and sex
*/
/*****************************************************************/
proc sort data=&data. out=&indicator._sorted;
        by &reporting_level. &sex. &equity_stratifier.;
        where &equity_stratifier. not in ('0', '', '.');
run;

* Case 1: Ordered data and the reference group is the highest group;
%if &ordered_data. eq yes and  &ordered_ref_group.=highest %then %do;
data &indicator.&yr._ref;
        set &indicator._sorted;
        by  &reporting_level. &sex. &equity_stratifier.;
        rename asr=asr_ref variance=variance_ref &equity_stratifier.=ref;
        if last.&sex. then output;
        keep &reporting_level. &sex. &equity_stratifier. asr variance;
run;
%end;

* Case 2: Ordered data and the reference group is the lowest group;
%if &ordered_data. eq yes and  &ordered_ref_group.=lowest %then %do;
data &indicator.&yr._ref;
        set &indicator._sorted;
        by  &reporting_level. &sex. &equity_stratifier.;
        rename asr=asr_ref variance=variance_ref &equity_stratifier.=ref;
        if first.&sex. then output;
        keep  &reporting_level. &sex. &equity_stratifier. asr variance;
run;
%end;

/* ERROR CONDITION: Unordered data and user has not specified whether the reference
   group is the 'highest' or 'lowest' value */
%if &ordered_data. eq yes and %length(&ordered_ref_group)=0 %then %do;
        %put 'ERROR: For ordered data, specify whether reference group is the 'highest'
                or 'lowest' group.';
        %ABORT;
```

```
%end;

* Case 3: For unordered data, the user must have specified the reference group;
%if &ordered_data. eq no and unordered_ref_group ne . %then %do;
data &indicator.&yr._ref;
	set &indicator._sorted;
	by &reporting_level. &sex.  &equity_stratifier.;
	rename asr=asr_ref variance=variance_ref &equity_stratifier.=ref;
	if &equity_stratifier. eq "&unordered_ref_group." then output;
	/* Might need to keep other outputs as well */
	keep  &reporting_level. &sex. &equity_stratifier. asr variance;
run;
%end;

* ERROR CONDITION: If unordered data and user has not specified the reference group;
%if &ordered_data. eq no and %length(&unordered_ref_group)=0 %then %do;
	%put 'ERROR: For unordered data, specify a character value for the reference group.';
	%ABORT;
%end;

data &indicator.&yr._ref;
	set &indicator.&yr._ref;
	label asr_ref    = "Age standardized rate (Reference group)"
	variance_ref = "Variance (Reference group)"
	ref ="Reference group";
run;

/****************************************************************/
/* STEP 2: Calculate rate ratio and rate difference and their confidence limits
/* for stratified data
/****************************************************************/
proc sort data=&data. out=&indicator._sorted2;
	by &reporting_level. &sex. &equity_stratifier.;
	where &equity_stratifier. not in ('', '.');
run;

/* Link dataset with reference group values to dataset with age standardized rates
	for all equity stratifier groups */
proc sql;
	create table &indicator._sorted3
	as select a.*, b.asr_ref as asr_ref, b.variance_ref as variance_ref, b.ref as ref
	from &indicator._sorted2 as a
	left join &indicator.&yr._ref as b
	on a.&reporting_level. = b.&reporting_level. and a.&sex. = b.&sex.;
```

```
quit;

/* Calculate rate ratio and rate difference and confidence limits */
data &outlib..rd_rr_&indicator._&yr._by_&equity_stratifier.;
	set &indicator._sorted3;
	/* Rate ratio calculations */
	if asr_ref ne . then rr=asr/asr_ref;
	else if asr_ref eq . then rr=.;

	if rr ne 0 and rr ne . then do;
		* Variance of the log of the rate ratio;
		var_logrr=((variance/(asr**2))+(variance_ref/(asr_ref**2)));
		* Upper and lower confidence limits of the rate ratio;
		lcl_rr = exp(log(rr)-(1.96*sqrt(var_logrr)));
		ucl_rr = exp(log(rr)+(1.96*sqrt(var_logrr)));
	end;

	if rr eq 0 or rr eq . then do;
			var_logrr=.;
		* Upper and lower confidence limits of the rate ratio;
			lcl_rr = .;
			ucl_rr = .;
	end;

	/* Rate difference calculations */
		if asr_ref ne . then rd=asr-asr_ref;
		else if asr_ref eq . then rd=.;

	if rd ne . then do;
		* Variance of the rate difference;
		var_rd=variance + variance_ref;
		* Upper and lower confidence limits of the rate difference;
		lcl_rd=rd-(1.96*sqrt(var_rd));
		ucl_rd=rd+(1.96*sqrt(var_rd));
	end;

	if rd eq . then do;
		* Variance of the rate difference;
		var_rd=.;
		* Upper and lower confidence limits of the rate difference;
		lcl_rd=.;
		ucl_rd=.;
	end;
```

```
	/* If reporting for the reference group the variance and CIs should not be reported */
	if &equity_stratifier. eq ref then do;
		var_logrr=.;
		lcl_rr = .;
		ucl_rr = .;
		var_rd=.;
		lcl_rd=.;
		ucl_rd=.;
	end;
run;

/* We do not report the rate difference and the rate ratio for the overall category ('0')
	but need to keep the overall category to join to PRR and PIN results in Step 4*/
data &outlib..rd_rr_&indicator._&yr._by_&equity_stratifier.;
	set &outlib..rd_rr_&indicator._&yr._by_&equity_stratifier.;
	if &equity_stratifier. eq '0' then do;
		rr=.;
		lcl_rr = .;
		ucl_rr = .;
		var_logrr=.;
		rd=.;
		lcl_rd=.;
		ucl_rd=.;
		var_rd=.;
	end;
run;

/*****************************************************************/
/* STEP 3: Calculate PRR and PIN */
/*****************************************************************/
/*****************************************************************/
/* 3A. CALCULATE POTENTIAL RATE REDUCTION (PRR)
	Requires:
	- Pi = proportion of the population in the ith category (by reporting level and sex)
	-- Use the total population (totalpop) to calculate the population proportion for each
			equity stratifier category by reporting level and sex
	- Rate ratios: (rate i/rate reference)
*/
/*****************************************************************/
* Calculate population proportion for each equity stratifier category by reporting level
	and sex;
proc sql;
	create table prr_&indicator.&yr._a as
	select &reporting_level., &sex., &equity_stratifier., totalpop, asr, variance,
```

```
        (totalpop/sum(totalpop)) as pop_portion
        from &indicator._sorted
        group by &reporting_level., &sex.;
quit;

* Create dataset containing population proportions and calculated rate ratios;
proc sql;
        create table prr_&indicator.&yr._b as
        select a.&reporting_level., a.&sex., a.&equity_stratifier., a.totalpop, a.asr, a.variance,
        a.pop_portion,          b.ref as ref, b.asr_ref as asr_ref, a.asr/asr_ref as rate_ratio
        from prr_&indicator.&yr._a as a inner join &indicator.&yr._ref as b
        on a.&reporting_level. = b.&reporting_level. and a.&sex. = b.&sex.;
quit;

* Calculate PRR by reporting level and sex;
proc sql;
        create table prr_&indicator.&yr. as
        select &reporting_level., &sex.,
        sum(pop_portion*(rate_ratio-1)) / (1+sum(pop_portion*(rate_ratio-1))) as prr
        from prr_&indicator.&yr._b
        group &reporting_level., &sex.;
quit;

/****************************************************************/
/* 3B. CALCULATE CONFIDENCE LIMITS FOR PRR
        Require:
        - Variance(rate ratio1:4,5) – for formulas see SAS macros and methodology notes
         */
/****************************************************************/
* i) Get values for comparison groups to calculate Sum(Pi*Pi*variance(rate))
        AND also Pi*rate (values for comparison groups are needed in subsequent
        calculations for CIs);
proc sql;
        Create table rate_comp_&indicator.&yr. as
        select &reporting_level., &sex., sum((pop_portion**2)*variance) as variance_comp,
        sum(pop_portion*asr) as sum_comp
        from prr_&indicator.&yr._b as a
        where &equity_stratifier. NE ref
        group by &reporting_level., &sex.
        order by &reporting_level., &sex., &equity_stratifier.;
quit;


* ii) Get values for reference group to be used in subsequent CI calculations;
```

```
proc sql;
	create table rate_ref_&indicator.&yr. as
	select &reporting_level., &sex., variance as variance_ref, asr as rate_ref,
	pop_portion as pop_portion_ref
	from prr_&indicator.&yr._b as a
	where &equity_stratifier. EQ ref
	order by &reporting_level., &sex., &equity_stratifier.;
quit;

* iii) Calculate upper and lower CIs for the PRR using the datasets created in step i and ii;
proc sql;
	create table prr_ci_&indicator.&yr. as
	select a.&reporting_level., a.&sex., prr,
	(sum_comp/rate_ref) as rate_comp,
	((variance_comp/sum_comp**2)+(variance_ref/rate_ref**2)) as varlog,
	exp(log(calculated rate_comp)-1.96*sqrt(calculated varlog)) as lcl_low,
	exp(log(calculated rate_comp)+1.96*sqrt(calculated varlog)) as ucl_low,
	1-(1/(pop_portion_ref+calculated lcl_low)) as lcl_prr,
	1-(1/(pop_portion_ref+calculated ucl_low)) as ucl_prr
	from prr_&indicator.&yr. as a, rate_comp_&indicator.&yr. as b,
	rate_ref_&indicator.&yr. as c
	where a.&reporting_level.=b.&reporting_level.=c.&reporting_level.
	and a.&sex.=b.&sex.=c.&sex.;
quit;

/*****************************************************************/
/* 3C. CREATE DATASET WITH PRR, confidence intervals and
		Population Impact Number (PIN) */
/*****************************************************************/
proc sql;
	create table prr_pin_&indicator.&yr._by_&equity_stratifier. as
	select &yr. as year, a.&reporting_level., a.&sex., a.&equity_stratifier.,
	prr, lcl_prr, ucl_prr, std_numer*prr as pin
	from &data. as a left join prr_ci_&indicator.&yr. as b
	on a.&reporting_level.=b.&reporting_level. and a.&sex.=b.&sex.;
quit;

* Report PRR and associated confidence intervals as a percentage;
data prr_pin_&indicator.&yr._by_&equity_stratifier.;
	set prr_pin_&indicator.&yr._by_&equity_stratifier.;
	prr=prr*100;
	lcl_prr=lcl_prr*100;
	ucl_prr=ucl_prr*100;
run;
```

```
proc sort data=prr_pin_&indicator.&yr._by_&equity_stratifier.
        out=&outlib..prr_pin_&indicator.&yr._by_&equity_stratifier. nodup;
        by &reporting_level. &sex. &equity_stratifier.;
run;

/* We only report the PRR and PIN for the overall category ('0') but need to
keep the other categories to join to RD and RR results in Step 4*/
data &outlib..prr_pin_&indicator.&yr._by_&equity_stratifier.;
        set &outlib..prr_pin_&indicator.&yr._by_&equity_stratifier.;
        if &equity_stratifier. ne '0' then do;
                prr=.;
                lcl_prr=.;
                ucl_prr=.;
                pin=.;
        end;
run;

/****************************************************************/
/* STEP 4: Output final data set with all rates and summary measures of inequality
*/
/****************************************************************/
proc sql;
        create table  &indicator._&equity_stratifier._inequalities as
        select a.year, a.&reporting_level.,  a.&equity_stratifier., a.&sex., a.count, a.totalpop,
        a.crude_rate as crude_rate, a.asr, a.asr_lci, a.asr_uci , a.rr, a.lcl_rr, a.ucl_rr, a.rd,
        a.lcl_rd, a.ucl_rd, b.prr, b.lcl_prr, b.ucl_prr, b.pin
        from &outlib..rd_rr_&indicator._&yr._by_&equity_stratifier. as a,
        &outlib..prr_pin_&indicator.&yr._by_&equity_stratifier. as b
        where a.&reporting_level.=b.&reporting_level. and a.&sex.=b.&sex.
        and a.&equity_stratifier. = b.&equity_stratifier.
        order by &reporting_level., &sex., &equity_stratifier.;
quit;


data &outlib..&outfile.;
        retain year &reporting_level. &equity_stratifier  &sex. count totalpop crude_rate asr
        asr_lci asr_uci rr lcl_rr ucl_rr rd         lcl_rd ucl_rd prr lcl_prr ucl_prr pin;
        length &equity_stratifier. $12.;
        set &indicator._&equity_stratifier._inequalities;

        label year = 'Year'
                &reporting_level. = 'Reporting level'
                &equity_stratifier. = "&equity_stratifier. "
```

```
&sex.            = 'Sex'
                 Count = 'Number of indicator cases'
                 totalpop = 'Total population'
                 crude_rate = "Crude rate per &MULTIPLIER.  population"
                 asr = "Age-standardized rate per &MULTIPLIER. population"
                 asr_lci  = 'Lower confidence limit of the age-standardized rate'
                 asr_uci= 'Upper confidence limit of the age-standardized rate'
                 rr = 'Rate Ratio (RR)'
                 lcl_rr = 'Lower confidence limit of RR'
                 ucl_rr = 'Upper confidence limit of RR'
                 rd = "Rate Difference (RD) per &MULTIPLIER.  population"
                 lcl_rd = 'Lower confidence limit of RD'
                 ucl_rd = 'Upper confidence limit of RD'
                 prr = 'Potential Rate Reduction (PRR) (%)'
                 lcl_prr = 'Lower confidence limit of PRR (%)'
                 ucl_prr = 'Upper confidence limit of PRR (%)'
                 pin = "Population Impact Number (PIN) per &MULTIPLIER. population";
    keep year &reporting_level. &equity_stratifier.  &sex. count totalpop crude_rate asr
    asr_lci asr_uci rr lcl_rr ucl_rr  rd lcl_rd ucl_rd prr lcl_prr ucl_prr pin;
run;

%mend;
```

Appendix D: R function — Calculate_Stratified_Rates

Output file

This function will produce a data set with crude and age-standardized rates for the chosen equity stratifier and reporting level.

Parameters

You must define all parameters listed below, except for parameters with default values, which can be defined as needed.

- yr — The year. Indicator cases and population estimates will be extracted only from the corresponding year. Only 1 year can be specified at a time.
- reporting_level — The geographic reporting level. The function calculates rates nationally and for the provinces and territories. Use the same variable name for reporting_level in your infile and popfile data sets.
- equity_stratifier — The variable for which you would like to calculate stratified rates (e.g., income quintile). This can be an ordered stratifier, such as income quintile, or an unordered stratifier, such as urban and rural/remote geographic location. You should use the same variable name for equity_stratifier in all your input data sets.
- infile — The name of the input data set of indicator cases.
- stdpop — The name of the standard population estimates data set.
- popfile — The name of the population estimates data set categorized by your equity stratifier (e.g., population estimates by income quintile).
- popvar — The population counts from popfile.
- age_var — The name of the age variable used consistently in the infile, stdpop and popfile data sets.
- agegpl — The lower age limit of the health indicator; must be in format 1, 2 . . . 14, 15 as specified in the age variable (e.g., if the lower age limit for your health indicator is 0 years, this value would be 1). Default value is 1.
- agegpu — The upper age limit of the health indicator; must be in format 1, 2 . . .18, 19 as specified in the age variable (e.g., if the upper age limit for your health indicator is 74 years, this value would be 15). Default value is 19.
- multiplier — The value for which you would like to present rates (e.g., multiplier = 100,000 for rates expressed per 100,000). Default value is 100,000.
- roundunit — The rounding unit (e.g., to round to 2 decimal places, specify ROUNDUNIT = 2). Default value is 2.

Function invocation

Calculate_Stratified_Rates (yr=, reporting_level=, equity_stratifier=, infile=, stdpop=, popfile=, popvar=, age_var=, agegpl=1, agegpu=19, multiplier=100000, roundunit=2)

The example invocation below represents how the function can be called if using the sample data provided in the Excel file.

Calculate_Stratified_Rates(yr=2020, reporting_level=PT_code, equity_stratifier=income_quintile, infile=Example_infile_data, stdpop=Example_stdpop_file, popfile=Example_popfile_data, popvar=PT_pop, age_var=AGE_GROUP_CODE, agegpl=1, agegpu=19, multiplier=100000, roundunit=2)

Function steps

1. Prepares the population denominators for the age range and year of interest. Creates overall population counts across equity stratifier levels.
2. Prepares indicator cases by creating aggregated counts for each combination of reporting level, age group, sex and equity stratifier. Creates additional overall counts across equity stratifier, sex and provincial/territorial levels (e.g., create national counts).
3. Prepares the standard population file used for age standardization for the age range of interest.
4. Merges the indicator cases and population count files.
5. Unmerges and transposes the indicator cases and population count files, so that each combination of sex, province/territory and equity stratifier is a column, and the rows correspond with each age group.
6. Calculates stratified and overall[iii] crude rates, and uses the following steps to calculate stratified and overall age-standardized rates:
    a. Calculates age-specific rates for each age group.
    b. Multiplies the age-specific rates of the population under study by the number of persons in each age group of the standard population to get the age-specific weighted rate for each age group.
    c. For each equity stratifier category, sums all age-specific weighted rates to get the age-standardized rate.
    d. Calculates variance and uses this to calculate confidence intervals.
7. Outputs the data table containing overall rates and rates by equity stratifier.

iii. Overall rates refer to all categories combined within the equity stratifier (e.g., for the income stratifier, this refers to the overall rate for quintiles 1 through 5). For this reason, any cases that are not assigned to an equity stratifier category (e.g., due to missing postal code) will be excluded from the overall rate.

Additional notes

This code **calls 2** external library packages: “dplyr”[3] and “reshape2.”[4]

Function code

```
"Calculate_Stratified_Rates" <-
 function (reporting_level, age_var, equity_stratifier, infile, popfile,
              popvar, stdpop, roundunit=2, multiplier=100000,
              agegpl=1, agegpu=19, yr)
 {

library(reshape2)
library(dplyr)

### rename popvar variable to PT_pop
   popfile <- popfile %>%
   rename(PT_pop=!!enquo(popvar))

### subset popfile and infile so only working with the required columns,
   # and remove any rows with missing values
   popfile <- popfile %>%
               select(POPULATION_YEAR, !!enquo(reporting_level), sex,
                        !!enquo(age_var), !!enquo(equity_stratifier),
                        PT_pop) %>%
               na.omit

  infile <- infile %>%
            select(YEAR, !!enquo(reporting_level), sex,
            !!enquo(age_var), !!enquo(equity_stratifier)) %>%
            na.omit

### format popfile

   #create overall counts by equity stratifier, where equity stratifier=0
   popfile_long_ES_agg <- aggregate(x = popfile[c("PT_pop")],
        by = popfile[c(deparse(substitute(reporting_level)),
        "sex", deparse(substitute(age_var)),
        "POPULATION_YEAR")], FUN = sum) %>%
        mutate(!!enquo(equity_stratifier) := "0") %>%
        bind_rows(popfile) %>% #combine with popfile
        mutate(pt_sex_es = paste(!!enquo(reporting_level), sex,
                                    !!enquo(equity_stratifier), sep = "_")) %>%
        filter (
```

```
        as.numeric(POPULATION_YEAR) == yr #only keep data from relevant year
       )

   #filter by age parameter
   popfile_long_ES_agg <-
      popfile_long_ES_agg[which(as.numeric(popfile_long_ES_agg
                                    [[deparse(substitute(age_var))]])
                                              >= agegpl
                                   & as.numeric(popfile_long_ES_agg
                                      [[deparse(substitute(age_var))]])
                                                     <= agegpu), ]

  #list population counts by pt_sex_es variable and age group
  popfile_tab <- aggregate(x = popfile_long_ES_agg[c("PT_pop")],
                           by = popfile_long_ES_agg[c("pt_sex_es",
                                 deparse(substitute(age_var)))],
                           FUN = max)

### format infile

  infile_long_sub <- infile %>%
     filter (as.numeric(YEAR) == yr) #only keep data from relevant year

  #filter by age parameter
  infile_long_sub <-
    infile_long_sub [which(as.numeric(infile_long_sub
                           [[deparse(substitute(age_var))]])
                                  >= agegpl
                          & as.numeric(infile_long_sub
                           [[deparse(substitute(age_var))]])
                                  <= agegpu), ]

  #create indicator counts by reporting level, sex, age, and
  #equity stratifier
  infile_tab <- infile_long_sub %>%
     count (!!enquo(reporting_level), sex, !!enquo(age_var),
              !!enquo(equity_stratifier), YEAR)

  #create overall counts by equity stratifier, where equity stratifier=0
  infile_tab_ES_agg <-
              aggregate(x = infile_tab[c("n")],
                        by = infile_tab
                        [c(deparse(substitute(reporting_level)),
                        "sex", deparse(substitute(age_var)), "YEAR")],
```

```
                          FUN = sum) %>%
    mutate(!!enquo(equity_stratifier) := "0") %>%
    bind_rows(infile_tab) #combine with the counts created in step above

  #create overall counts by sex, where sex=3
  infile_tab_ES_sex_agg <- aggregate(x = infile_tab_ES_agg[c("n")],
    by = infile_tab_ES_agg[c(deparse(substitute(reporting_level)),
                             deparse(substitute(equity_stratifier)),
                             deparse(substitute(age_var)), "YEAR")],
     FUN = sum) %>%
    mutate(sex = "3") %>%
    bind_rows(infile_tab_ES_agg) #combine with counts created in step above

  #create overall counts by reporting level, reporting level=99
  infile_tab_ES_sex_PT_agg <-
        aggregate(x = infile_tab_ES_sex_agg[c("n")],
                  by = infile_tab_ES_sex_agg[c("sex",
                        deparse(substitute(equity_stratifier)),
                        deparse(substitute(age_var)), "YEAR")],
                  FUN = sum)%>%
    mutate(!!enquo(reporting_level) := "99") %>%
    bind_rows(infile_tab_ES_sex_agg) %>% #combine with counts created above
    mutate(pt_sex_es = paste(!!enquo(reporting_level), sex,
                             !!enquo(equity_stratifier), sep="_")) %>%
    select(!!enquo(age_var), pt_sex_es, n)
     #list indicator counts by pt_sex_es variable and age group



### Merge the indicator and population files, so that they match
  # by pt_sex_es, and age group. Then remove any combinations of PT*sex*ES in
  # indicator cases that do not have a corresponding population count.
  # For combinations of PT*sex*ES in population counts that do not have a
  # corresponding indicator case, set number of indicator cases to zero

  merged_pop_infile <- merge(popfile_tab,  infile_tab_ES_sex_PT_agg ,
                             by.x=c("pt_sex_es",
                                    deparse(substitute(age_var))),
                             by.y=c("pt_sex_es",
                                    deparse(substitute(age_var))),
                              all.x=TRUE) %>%
                       mutate(n=ifelse(is.na(n), 0, n))

### Unmerge the indicator and population file at the end,
```

```
  # so they can be transformed into format needed for
  # "calc.age.adjusted.rates"

  pop_data <- recast( merged_pop_infile[,c('pt_sex_es',
                                deparse(substitute(age_var)), 'PT_pop')],
                          formula(paste(substitute(age_var), "+",
                            substitute(variable),
                            "~", substitute(pt_sex_es))),
                        id.var = c("pt_sex_es",
                                 deparse(substitute(age_var))))
                        #change to a wide format

  #order from lowest to highest age group
  pop_data <-pop_data[order(
   as.numeric(pop_data[[deparse(substitute(age_var))]])), ]

  pop_data[,-c(1,2)][pop_data[,-c(1,2)]==0] <- NA
   #set any 0 population counts to NA so calculations can skip the NA's

  count_data <-
   recast( merged_pop_infile[,c('pt_sex_es',
              deparse(substitute(age_var)), 'n')],
              formula(paste(substitute(age_var), "+",
                            substitute(variable), "~",
                            substitute(pt_sex_es))),
              id.var = c("pt_sex_es", deparse(substitute(age_var))))
              #change to a wide format

  #order from lowest to highest age group
  count_data <-
       count_data[order(as.numeric
                 (count_data[[deparse(substitute(age_var))]])), ]

####format stdpop to stdpop_list

  #filter by age parameter
  stdpop_data_sub  <-
         stdpop[which(as.numeric(
          stdpop[[deparse(substitute(age_var))]]) >= agegpl
         & as.numeric(
          stdpop[[deparse(substitute(age_var))]]) <= agegpu), ]

  stdpop_list<- stdpop_data_sub$STANDARD_POP_CNT
  names(stdpop_list) <- stdpop_data_sub[[deparse(substitute(age_var))]]
```

```
    stdpop_list <- stdpop_list[order(as.numeric(names(stdpop_list)))]
    #order from lowest to highest age group


  ###function to create direct age adjusted stratified rates for each
    #combo of sex*P/T*Equity stratifier

    "calc.age.adjusted.rates" <-
     function (count, pop, stdpop)
     {
      if(missing(pop))
        pop <- numeric(0)
      rate <- count/pop
      cruderate <- sum(count, na.rm = TRUE)/sum(pop,na.rm = TRUE)
      stdwt <- stdpop/sum(stdpop)
      dsr <- sum(stdwt * rate, na.rm = TRUE)
      variance=sum((stdwt^2)*(count/pop*multiplier)*
                           ((multiplier-(count/pop*multiplier))/pop), na.rm=TRUE)
      lci <- dsr*multiplier - 1.96*sqrt(variance)
      uci <- dsr*multiplier + 1.96*sqrt(variance)
      if (lci < 0 ) {
       p <- dsr
       if (p==0) {
         p <- 1/(multiplier^2)}
       logitp <- log(p/(1-p))
       var <- 1/(p*p*(1-p)*(1-p))*(variance/(multiplier^2))
       A <- logitp-1.96*sqrt(var)
       B <- logitp+1.96*sqrt(var)
       lci <- multiplier/(1+exp(-A))
       uci <- multiplier/(1+exp(-B))
      }

      c(year=yr,
        PT_code = strsplit(colnames(count), "_")[[1]][1],
        sex = strsplit(colnames(count), "_")[[1]][2],
        Equity_stratifier = strsplit(colnames(count), "_")[[1]][3],
        number_indicator_cases = sum(count, na.rm=TRUE),
        totalpop = sum(pop, na.rm=TRUE),
        crude_rate = round(cruderate*multiplier, roundunit),
        variance=round(variance, roundunit),
        asr = round(dsr*multiplier, roundunit),
        asr_lci = round(lci, roundunit),
        asr_uci = round(uci, roundunit),
        std_numer = round(sum(count/pop*stdpop, na.rm=TRUE), roundunit))
```

```
    }


###apply the function across the prepared data frames

  final <- data.frame(matrix(NA, nrow = 0, ncol = 12))

  for (i in 3:ncol(count_data)) {
    res <- c(calc.age.adjusted.rates(count_data[i],
                                     pop_data[i],stdpop=stdpop_list))
    final[nrow(final) + 1,] <- res
  }


  colnames(final) <- c("year", deparse(substitute(reporting_level)),
                       "sex", deparse(substitute(equity_stratifier)),
                       "count", "totalpop", "crude_rate", "variance",
                       "asr", "asr_lci", "asr_uci", "std_numer")


### set all columns as numeric values, except for stratifier columns
    final <- as.data.frame(lapply(final,as.numeric))

    cols.char <- c(deparse(substitute(reporting_level)),
                   "sex", deparse(substitute(equity_stratifier)))
    final[cols.char] <- lapply(final[cols.char],as.character)


### output data
final

 }
```

Appendix E: R function — Calculate_Inequality_Measures

Output file

This function will produce a data set with final RR, RD, PRR and PIN results based on equity stratifier and reporting level.

Parameters

You must define all parameters listed below.

- *data* — The name of the data set being input into the function.
- *equity_stratifier* — The variable for which you would like to calculate inequality measures (e.g., income quintile).
- *ref_group* —You must specify the reference group. This is a numeric value representing the group that all other groups are compared against. It cannot be specified as 0, as this value is reserved for calculating overall rates.
- *reporting_level* — The geographic reporting level.
- *sex* — The sex variable for the indicator. It must be included even if reporting for both sexes (use a dummy in its place).

Function invocation

Calculate_Inequality_Measures (data=, equity_stratifier=, ref_group=, reporting_level=, sex=)

The example invocation below represents how the function can be called if using the output created from running the fake sample data provided in the Excel file through Calculate_Stratified_Rates.

Calculate_Inequality_Measures(data=fakedata_output1, equity_stratifier=income_quintile, ref_group=5, reporting_level=PT_code, sex=sex)

Function steps

1. Creates a data set containing the reference group for each combination of reporting_level and sex.
2. Calculates RR and RD.
3. Calculates PRR and PIN.
4. Outputs data tables with all summary measures of inequality.

Additional notes

- This code **calls 1** external library package: "dplyr."[3]
- The presented crude and age standardized rates, RD and PIN produced in the output of this function should be interpreted per your chosen multiplier for the input data set (e.g., if the input data set age standardized rates were calculated per 100,000, then the PIN should be interpreted as per 100,000 as well).

Function code

```
"Calculate_Inequality_Measures" <- function (data, equity_stratifier,
                                              ref_group, reporting_level, sex)
{
 library(dplyr)

###will produce error is user specifies 0 as the reference group
 if(ref_group == 0) stop(
  'ref_group cannot be specified as 0. Value 0 is reserved for overall rate')


###rename parameter variables
    data <- data %>%
     rename(reporting_var=!!enquo(reporting_level)) %>%
     rename(equity_var=!!enquo(equity_stratifier))  %>%
     rename(sex_var = !!enquo(sex))

###extract asr and variance of the reference level
    refgroups <- data %>% filter(equity_var==paste(ref_group)) %>%
               mutate (asr_ref = asr, variance_ref = variance,
                      ref=ref_group) %>%
               select(year, reporting_var, sex_var, asr_ref,
                     variance_ref, ref)

###calculate RR and RR confidence intervals
   #(except for when equity stratifier is 0).
   #Don't calculate confidence intervals if rr=0 or NA
     data_rr <- merge(data, refgroups,
                      by=c("sex_var", "reporting_var", "year"))  %>%
               mutate(rr = ifelse(equity_var=='0', NA, asr/asr_ref)) %>%
               mutate(var_logrr = ifelse(!is.na(rr) |rr != 0 ,
                        (variance/(asr**2))+ (variance_ref/
                                         (asr_ref**2)),
                        NA)) %>%
```

```
                    mutate(lcl_rr = ifelse(!is.na(rr) |rr != 0 ,
                                           exp(log(rr)-(1.96*sqrt(var_logrr))),
                                           NA)) %>%
                    mutate(ucl_rr = ifelse(!is.na(rr) |rr != 0 ,
                                           exp(log(rr)+(1.96*sqrt(var_logrr))),
                                           NA))

###calculate RD and RD confidence intervals
  #(except for when equity stratifier is 0).
  #Don't calculate confidence intervals if rd=NA
  data_rd <- data_rr %>%
          mutate(rd = ifelse(equity_var=='0', NA, asr-asr_ref)) %>%
          mutate(var_rd = ifelse(!is.na(rd),
                                 variance+variance_ref,
                                 NA)) %>%
          mutate(lcl_rd = ifelse(!is.na(rd),
                                 rd-(1.96*sqrt(var_rd)),
                                 NA)) %>%
          mutate(ucl_rd = ifelse(!is.na(rd),
                                 rd+(1.96*sqrt(var_rd)),
                                 NA)) %>%
  #set all variance or CI estimates for the reference levels to NA
    mutate(
     var_logrr=replace(var_logrr, equity_var == as.character(ref), NA),
     lcl_rr=replace(lcl_rr, equity_var == as.character(ref), NA),
     ucl_rr=replace(ucl_rr, equity_var == as.character(ref), NA),
     var_rd=replace(var_rd, equity_var == as.character(ref), NA),
     lcl_rd=replace(lcl_rd, equity_var == as.character(ref), NA),
     ucl_rd=replace(ucl_rd, equity_var == as.character(ref), NA))


###Calculations for PRR

  data_prr <- data_rd %>%
    #Calculate population proportion for each equity stratifier category
     # by reporting level and sex
    filter (equity_var !="0") %>%
    group_by(sex_var, reporting_var) %>%
    mutate(totalpop_sex_var_geo = sum(totalpop, na.rm = TRUE)) %>%
    ungroup() %>%
    mutate(pop_portion = totalpop/totalpop_sex_var_geo) %>%
    group_by(sex_var, reporting_var) %>%
    #calculate prr
      mutate(prr = (sum(pop_portion*(rr-1), na.rm=TRUE) /
```

```
                        (1+sum(pop_portion*(rr-1),na.rm=TRUE)))) %>%
  #copy the prr value into each row of equity stratifier variables
    mutate(pop_portion_ref = ifelse(equity_var==as.character(ref),
                                    pop_portion, NA) ) %>%
    mutate(pop_portion_ref =
              pop_portion_ref[which(equity_var==as.character(ref))])

 #get values for comparison groups (need for subsequent calculations)
   data_prr_comp <- data_prr %>%
                        filter (equity_var!=as.character(ref)) %>%
                        group_by(sex_var, reporting_var) %>%
                        mutate(variance_comp = sum((pop_portion**2)*variance,
                                                   na.rm=TRUE),
                             sum_comp = sum(pop_portion*asr, na.rm=TRUE))

 # calculate prr confidence intervals
 data_prr_cl  <- data_prr_comp %>%
   mutate(rate_comp = sum_comp/asr_ref,
          varlog = ((variance_comp/sum_comp**2)+(variance_ref/
                                                   asr_ref**2)),
          lcl_low = exp(log(rate_comp)-1.96*sqrt(varlog)),
          ucl_low = exp(log(rate_comp)+1.96*sqrt(varlog)),
          lcl_prr = 1-(1/(pop_portion_ref+lcl_low)),
          ucl_prr = 1-(1/(pop_portion_ref+ucl_low)),
          prr = prr*100,
          lcl_prr = lcl_prr*100,
          ucl_prr = ucl_prr*100) %>%
          select(year, sex_var, reporting_var, equity_var, prr,
              lcl_prr, ucl_prr) %>%
          mutate(equity_var = '0') %>%
          distinct()
 #resulting dataset with one row for each reporting level and sex,
 #  equity_var=0

#merge the rd and rr results with the prr results for final output table
 data_output <- merge(data_rd, data_prr_cl,
                         by=c("sex_var", "reporting_var", "year",
                             "equity_var"), all.x = TRUE) %>%
                    mutate(pin = prr*std_numer/100) %>% #calculate PIN
                    select(year,
                        reporting_var, # Reporting level
                        equity_var,    # Equity stratifier
                        sex_var,       # Sex
                        count,         # Number of indicator cases
```

```
                              totalpop,     # Total population
                              crude_rate,   # Crude rate per your
                                            #   multipliers population
                              asr,          # Age-standardized rate per
                                            #   your multipliers population
                              asr_lci, # Lower confidence limit of the asr
                              asr_uci, # Upper confidence limit of the asr
                              rr,     # Rate Ratio
                              lcl_rr,  # Lower confidence limit of RR
                              ucl_rr,  # Upper confidence limit of RR
                              rd,     # Rate Difference per your
                                      # multipliers population
                              lcl_rd,  # Lower confidence limit of RD
                              ucl_rd,  # Upper confidence limit of RD
                              prr,    # Potential Rate Reduction (PRR) (%)
                              lcl_prr, # Lower confidence limit of PRR (%)
                              ucl_prr, # Upper confidence limit of PRR (%)
                              pin) %>% # Population Impact Number (PIN)
                                       # per your multipliers population
          arrange(reporting_var, sex_var, equity_var) %>%
          rename(!!enquo(reporting_level):=reporting_var) %>%
          rename(!!enquo(equity_stratifier):=equity_var)  %>%
          rename(!!enquo(sex):=sex_var )


  data_output

  }
```

References

1. Rothman K, Greenland S, Lash T. *Modern Epidemiology*. 2008.

2. Rockhill B, Newman B, Weinberg C. Use and misuse of population attributable fractions. *American Journal of Public Health*. January 1998.

3. Wickham H et al. dplyr: A Grammar of Data Manipulation — R package version 1.0.7. 2021.

4. Wickham H. Reshaping data with the reshape package. *Journal of Statistical Software*. 21(12), 1–20. 2007.

help@cihi.ca

CIHI Ottawa
495 Richmond Road
Suite 600
Ottawa, Ont.
K2A 4H6
613-241-7860

CIHI Toronto
4110 Yonge Street
Suite 300
Toronto, Ont.
M2P 2B7
416-481-2002

CIHI Victoria
880 Douglas Street
Suite 600
Victoria, B.C.
V8W 2B7
250-220-4100

CIHI Montréal
1010 Sherbrooke Street West
Suite 602
Montréal, Que.
H3A 2R7
514-842-2226

cihi.ca

10152-0522