Within and Across Group Variations in Ambient Climate Exposures Across Socio-Demographic Groups

Fan Wang

2023-01-01

Contents

1 Distribution of Ambient Climate Exposures Across and Within Socio-Demographic Groups

Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools research support package, R4Econ examples page, PkgTestR packaging guide, or Stat4Econ course page.

1.1 Location, Population, and Environmental Exposures (Pollution)

1.1.1 Local Public Exposures $Z_{l,y}$

Environmental exposure, specifically PM 2.5, is a local public good (bad). Prior research has shown that due to the nature of particular matter formation in the air, while there is particular matter variation at the city level, there are no significant variations in particular matter pollution at the neighborhood level He et al. (2019) and Liu et al. (2022). Additionally, there is a difference between ambient environmental exposures

and the amount of particular matter inhaled by individual residents: the former is shared by all residents, but the latter can differ depending on individuals' physical and socio-economic attributes.

There are M different locations (cities/counties/townships), indexed from $l = 1$ to $l = M$. In a particular time-period, we assume identical potential pollution exposure for all residents in the same location. Suppose additionally that we compute statistics at the interval of period y (which we will call years), and each period includes sub-period (which we will call daily) pollution measure for each sub-period t, indexed from $t = 1$ to $t = T_y$, where T_y is the total number of sub-periods within a period.

We assume now additionally that individuals stay in the same location during the course of period y (year). Let $\mathrm{Z}_{l,y}$ be the total (potential) pollution exposure by a resident in location l during period y. This is equal to the sum of (potential) pollution exposure during the course of a period:

$$Z_{l,y} = \sum_{t=1}^{T_y} Z_{l,y,t}$$

Note that: - "Same location during period" is assumed due to data limitation. If we know the residential location patterns of individuals during the course of period y, we can adjust the sum above so that location-specific pollution data is only added for the sub-period in which the individual reside at the location. - "Common exposure within location" can be an innocuous assumption depending on the precision of pollution monitoring stations.

1.1.2 Group Exposures $\mathcal{Z}_{i,y}$

Let there be N population groups, indexed from $i = 1$ to $i = N$. The N population groups reside in the M locations. Let $P_{l,i,y}$ denote share of overall population belonging to group i that resides in location l during period y:

$$1 = \sum_{l=1}^{M} \sum_{i=1}^{N} P_{l,i,y} \text{ for all } y$$

Note that population groups can be defined by any characteristics, including location attributes. When M are demographic (gender, age, etc) groups, we can study the differences in pollution distribution across demographic groups. When M are locations, we can study how pollution distributional burdens vary overall and vary with and across regions.

Suppose M represents M villages and N is the number of counties which the villages belong to. Each village can only belong to one county, then $P_{l,i,y}$, where l is the village indicator and i is the county indicator, has the property that:

$$\text{If } P_{l,i,y} > 0, \text{ then } P_{l,\widehat{i},y} = 0, \text{ for all } \widehat{i} \neq i$$

This means that l is not a sub-index for i. Think of l as the row index, and i as the column index. There are M rows and N columns.

The share of overall population that belongs to location l is:

$$P_{l,y} = \sum_{i=1}^{N} P_{l,i,y}$$

And the share of overall population that belongs to population group i is:

$$\mathcal{P}_{i,y} = \sum_{l=1}^{M} P_{l,i,y}$$

$\mathcal{Z}_{i,y}$, which is the average pollution exposure facing an individual belonging to group i in year y, is determined by how individuals from population group i are distributed across the M locations:

$$\mathcal{Z}_{i,y} = \sum_{l=1}^{M} \frac{P_{l,i,y}}{\mathcal{P}_{i,y}} \times Z_{l,y}$$

Additionally, Z_y is the average pollution exposure facing an individual, regardless of population group, in year y:

$$Z_y = \sum_{i=1}^{N} (\mathcal{P}_{i,y} \times \mathcal{Z}_{i,y})$$

Note that we use $Z_{l,y}$ and $P_{l,y}$ for location-specific information, and calligraphic $\mathcal{Z}_{i,y}$ and $\mathcal{P}_{i,y}$ for population-group-specific information.

1.1.3 Excess Environmental Exposure Burden to Population

We have $\mathcal{Z}_{i,y}$, the average pollution burden facing an individual in a particular population group. We also have P_i, the share of population belong to population group i.

How does the share of pollution burden facing a population group relate to the share of population this group has in the overall population?

We define $\mathcal{E}_{i,y}$ as the share of pollution burden for population group i that is in excess of its population share as:

$$\mathcal{E}_{i,y} = \left(\left(\frac{\mathcal{P}_{i,y} \times \mathcal{Z}_{i,y}}{\sum_{\widehat{i}}^{N} \left(\mathcal{P}_{\widehat{i},y} \times \mathcal{Z}_{\widehat{i},y} \right)} \right) \times \frac{1}{\mathcal{P}_{i,y}} \right) - 1 = \frac{\mathcal{Z}_{i,y}}{Z_y} - 1$$

Note that $\mathcal{E}_{i,y}$ is simply the ratio between the average pollution exposure for an individual in group i in year y and the overall average pollution exposure for an individual in year y, minus 1.

1.1.4 Exposure Distribution Percentiles

We are interested in distributional statistics: how much (potential) pollution exposure inequality is there? This could be computed in three ways: 1. overall; 2. across groups; 3. within groups. Let $Q_x(\tau)$ be the quantile function for random variable x where $\tau \in [0, 1)$ and $\tau = 0.01$ is 1 percent. We have separate quantile functions for each of the three cases:

1. Overall inequality in pollution exposure across all population. Since all differences in pollution exposure comes from variations in pollution across the M locations, this will be measuring inequality across all locations:

$$\left\{ P_{l,y} = \left(\sum_{i=1}^{N} P_{l,i,y} \right), Z_{l,y} \right\}_{l=1}^{M}$$

$$Q_{Z_y}(\tau) = \max_{l \in \{1,\ldots,M\}} \left(\mathrm{Z}_{l,y} \times \mathbf{1} \left\{ \tau \geq \sum_{\tilde{l}=1}^{M} P_{\tilde{l},y} \times \mathbf{1} \left\{ Z_{\tilde{l},y} \leq Z_{l,y} \right\} \right\} \right)$$

2. Distributional differences across groups:

$$\left\{ \mathcal{P}_{i,y} = \left(\sum_{l=1}^{M} P_{l,i,y} \right), \mathcal{Z}_{i,y} = \left(\sum_{l=1}^{M} \frac{P_{l,i,y}}{\mathcal{P}_{i,y}} \times Z_{l,y} \right) \right\}_{i=1}^{N}$$

$$Q_{\mathcal{Z}_y}(\tau) = \max_{i \in \{1,\ldots,N\}} \left(\mathcal{Z}_{i,y} \times \mathbf{1} \left\{ \tau \geq \sum_{\tilde{i}=1}^{N} \mathcal{P}_{\tilde{i},y} \times \mathbf{1} \left\{ \mathcal{Z}_{\tilde{i},y} \leq \mathcal{Z}_{i,y} \right\} \right\} \right)$$

3. Distributional differences within groups. Meaning that we can compute N sets of distributional statistics based on $\left\{\frac{P_{l,i,y}}{\mathcal{P}_{i,y}}, Z_{l,y}\right\}_{l=1}^{M}$ from $i = 1$ to N. We have:

$$Q_{Z_{i,y}}(\tau) = \max_{l \in \{1,\ldots,M\}} \left(\mathrm{Z}_{l,y} \times \mathbf{1}\left\{ \tau \geq \sum_{\tilde{l}=1}^{M} \frac{P_{\tilde{l},i,y}}{\mathcal{P}_{i,y}} \times \mathbf{1}\left\{ Z_{\tilde{l},y} \leq Z_{l,y} \right\} \right\} \right), \forall i \in \{1,\ldots,N\}$$

Note that the $Q_{_{i,y}}(\tau)$ differs for each i because of the i-specific population distribution across location differs ($\frac{P_{\tilde{l},i,y}}{\mathcal{P}_{i,y}}$). But all type 3 within group (and type 1 overall) quantile functions share the same pollution data $Z_{l,y}$. These quantile functions are step-functions. Each consecutive and rising step, in 2-d, has "run" and "rise". Type 1 and 3 statistics share the same "run" but different "rise". Type 2 statistics have differing runs and rises.

Given $Q_x(\tau)$, we can construct the ratio of two quantiles. Of particular interest is the ratio of the 80th and the 20th quantile:

$$\mathcal{Q}_x^{80/20} = \frac{Q_x(\tau = 0.8)}{Q_x(\tau = 0.2)}$$

Where we use $\mathcal{Q}_x^{80/20}$ to denote the P80 to P20 ratio for random variable x. We can compare this statistics across population groups to see which group has larger within group environmental exposure variation. We can also compare statistics from the overall distribution, the across group distribution, and the within group distributions. Note that Type (1) overall P80 to P20 ratio must be greater or equal to Type (2) across group P80 to P20 ratio:

$$\mathcal{Q}_{Z_y}^{80/20} \geq \mathcal{Q}_{\mathcal{Z}_y}^{80/20}$$

1.1.5 Gini, Atkinson, and S.D. Functions

In addition to percentile ratios, we can also compute GINI, Atkinson, and variance based statistics overall, across groups, and within groups. The percentile based results are potentially easier to interpret. They are in effect ratios of relative excess burdens.

For variations overall and within/across groups, note that the GINI and ATKINSON require positive values for inputs, so they can be used with $\mathcal{Z}_{i,y}$, but they can not be applied to the $\mathcal{E}_{i,y}$, which is positive or negative. For variations across groups, we compute these statistics given the distribution of P_i.

First, we generate an internal function for GINI, given some sorted array $\mathbf{x}$:

$$\text{GINI Index} = 1 - \frac{2}{N+1} \cdot \left(\sum_{i=1}^{N} \sum_{j=1}^{i} x_j \right) \cdot \left(\sum_{i=1}^{N} x_i \right)^{-1}$$

We use the Gini index which is documented in on the fs_gini_disc page.

```
ffi_dist_gini_random_var_pos_test <- function(ar_data, ar_prob_data) {
  #' @param ar_data array sorted array values low to high
  #' @param ar_prob_data array probability mass for each element along `ar_data`, sums to 1

  fl_mean <- sum(ar_data*ar_prob_data);
  ar_mean_cumsum <- cumsum(ar_data*ar_prob_data);
  ar_height <- ar_mean_cumsum/fl_mean;
  fl_area_drm <- sum(ar_prob_data*ar_height);
  fl_area_below45 <- sum(ar_prob_data*(cumsum(ar_prob_data)/sum(ar_prob_data)))
  fl_gini_index <- (fl_area_below45-fl_area_drm)/fl_area_below45
  return(fl_gini_index)
}
```

Second, we create an internal function for Atkinson statistics, from Atkinson (JET, 1970). The formula is:

$$\text{Atkinson Index} = A\left(\{Y_i\}_{i=1}^{N}, \lambda\right) = 1 - \left(\sum_{i=1}^{N} \frac{1}{N}\left(\frac{Y_i}{\sum_{j=1}^{N}\left(\frac{Y_j}{N}\right)}\right)^{\lambda}\right)^{\frac{1}{\lambda}} \in [0,1]$$

Note that the Atkinson statistics differ by planner preference, and is arbitrary. We use the Atkinson index formula which is documented on the fs_atkinson_ces page.

```
# Formula
ffi_atkinson_random_var_ineq <- function(ar_data, ar_prob_data, fl_rho) {
  #' @param ar_data array sorted array values
  #' @param ar_prob_data array probability mass for each element along `ar_data`, sums to 1
  #' @param fl_rho float inequality aversion parameter fl_rho = 1 for planner
  #' without inequality aversion. fl_rho = -infinity for fully inequality averse.

  fl_mean <- sum(ar_data*ar_prob_data);
  fl_atkinson <- 1 - (sum(ar_prob_data*(ar_data^{fl_rho}))^(1/fl_rho))/fl_mean
  return(fl_atkinson)
}
```

Third, we also create an internal function for standard deviation for discrete random variable. This function generates the standard deviation as well as the coefficient of variation.

```
# Formula
ffi_std_cov <- function(ar_data, ar_prob_data) {
  #' @param ar_data array array values
  #' @param ar_prob_data array probability mass for each element along `ar_data`, sums to 1

  fl_mean <- sum(ar_data*ar_prob_data)
  fl_std <- sqrt(sum(ar_prob_data*(ar_data - fl_mean)^2))
  fl_coef_of_variation <- fl_std/fl_mean

  ls_fl_std_cov <- list("std"=fl_std, "cov"= fl_coef_of_variation, "mean" = fl_mean)

  return(ls_fl_std_cov)
}
```

1.1.6 Scenarios for Within Group Exposure Variations

The prior sections discussed how to measure environmental exposure (pollution) variations across groups. For example, we might learn that $\mathcal{E}_{i=\text{white},y} = -0.60$ and $\mathcal{E}_{i=\text{black},y} = +0.60$: black population is a lot more exposed to pollution relative to their population than white people are, due to where they live.

The next question we want to ask is how much variation there is in environmental exposure within group. The interest in this is natural. Limited difference in across group variation could mask great inequality in environmental exposures within group, something relevant for the concept of environmental justice. Additionally, if variations within group are large, it is not clear that mean difference across group would properly capture the magnitude of environmental inequality that exists within a population.

Suppose there are two locations and two population groups, we have the following scenarios:

1. Zero within group variation for both population groups:
   - In the most extreme case, all people of one population group live in one location, and all people of another population group live in another location.
   - In this case, there would be zero within group variation in environmental exposure.
2. Positive within group variation, but identical within group variation for both population groups:

- Suppose the distribution of each of the population groups across locations could be identical (say 40 percent in location 1, 60 percent in location 2).
- As long as environmental exposure differs across the locations, there will be within group exposure variation.
- However, the within group variations will be the same for both population groups.

3. Different within group variations across population groups:
   - The population distribution across location varies for the population groups. Group 1 might have 40 percent in location 1, group 2 might have 70 percent in location 1.
   - Same as case (2), as long as environmental exposure differs across the locations, there will be within group exposure variation.
   - But now, additionally, the within group variations will be different across the population groups. In the most extreme case, one group—if they all reside in one location—might have no within group variation, however, another group—if they co-resides in multiple locations—will have positive within group variations.

These three scenarios above present contrasting worlds of environmental inequality which can only be captured by summarizing the magnitude of within-group exposure inequalities.

1.2 Simulate Population Distribution over Location and Demographics

Use the binomial distribution to generate heterogenous demographic break-down by location. There are N demographic cells, and the binomial distribution provides the probability mass in each of the N cell. Different bernoulli "win" chance for each location. There is also probability distribution over population in each location.

First, construct empty population share dataframe:

```
# Percentiles to be used for overall, across group, as well as within group calculations.
# For later purposes, what are the percentiles of interest to compute
ar_fl_percentiles <- c(0.1, 0.2, 0.8, 0.9)
# For Percentile ratios of interest, specify lower and upper bounds
ar_fl_ratio_upper <- c(0.8, 0.9)
ar_fl_ratio_lower <- c(0.2, 0.1)

# 7 different age groups and 12 different locationso
it_N_pop_groups <- 100
it_M_location <- 20
# it_N_pop_groups <- 7
# it_M_location <- 10
# Matrix of demographics by location
mt_pop_data_frac <- matrix(data=NA, nrow=it_M_location, ncol=it_N_pop_groups)
colnames(mt_pop_data_frac) <- paste0('popgrp', seq(1,it_N_pop_groups))
rownames(mt_pop_data_frac) <- paste0('location', seq(1,it_M_location))

# For succinct visualization select subset of population groups to display
it_popgrp_disp <- 7
ar_it_popgrp_disp <- seq(1, it_N_pop_groups, length.out=it_popgrp_disp)
ar_it_popgrp_disp <- round(ar_it_popgrp_disp)
ar_it_popgrp_disp_withoverall <- c(ar_it_popgrp_disp, it_N_pop_groups+1, it_N_pop_groups+2)
st_popgrp_disp <- paste0('(', it_popgrp_disp, ' of ', it_N_pop_groups, ' pop-groups shown)')
it_loc_disp <- 10
ar_it_loc_disp <- seq(1, it_M_location, length.out=it_loc_disp)
ar_it_loc_disp <- round(ar_it_loc_disp)
st_loc_disp <- paste0('(', it_loc_disp, ' of ', it_M_location, ' locations shown)')

# Display
```

```
st_caption = paste('Location and demographic cell',
  st_popgrp_disp, st_loc_disp, sep=" ")
mt_pop_data_frac[ar_it_loc_disp, ar_it_popgrp_disp] %>%
  kable(caption = st_caption) %>%
  kable_styling_fc()
```

Location and demographic cell (7 of 100 pop-groups shown) (10 of 20 locations shown)

	popgrp1	popgrp18	popgrp34	popgrp50	popgrp67	popgrp84	popgrp100
location1	NA	NA	NA	NA	NA	NA	NA
location3	NA	NA	NA	NA	NA	NA	NA
location5	NA	NA	NA	NA	NA	NA	NA
location7	NA	NA	NA	NA	NA	NA	NA
location9	NA	NA	NA	NA	NA	NA	NA
location12	NA	NA	NA	NA	NA	NA	NA
location14	NA	NA	NA	NA	NA	NA	NA
location16	NA	NA	NA	NA	NA	NA	NA
location18	NA	NA	NA	NA	NA	NA	NA
location20	NA	NA	NA	NA	NA	NA	NA

Second, generate conditional population distribution for each location, and then multiply by the share of population in each locality:

```
# Share of population per location
set.seed(123)
ar_p_loc <- dbinom(0:(3*it_M_location-1), 3*it_M_location-1, 0.5)
it_start <- length(ar_p_loc)/2-it_M_location/2
ar_p_loc <- ar_p_loc[it_start:(it_start+it_M_location-1)]
ar_p_loc <- ar_p_loc/sum(ar_p_loc)

# Different bernoulli "win" probability for each location
set.seed(234)
# ar_fl_unif_prob <- sort(runif(it_M_location)*(0.25)+0.4)
ar_fl_unif_prob <- sort(runif(it_M_location))

# Generate population proportion by locality
for (it_loc in 1:it_M_location ) {
  ar_p_pop_condi_loc <- dbinom(0:(it_N_pop_groups-1), it_N_pop_groups-1, ar_fl_unif_prob[it_loc])
  mt_pop_data_frac[it_loc,] <- ar_p_pop_condi_loc*ar_p_loc[it_loc]
}

# Sum of cells, should equal to 1
print(paste0('pop frac sum = ', sum(mt_pop_data_frac)))
```

```
## [1] "pop frac sum = 1"
```

```
# Display
st_caption = paste('Share of population in each location and demographic cell',
  st_popgrp_disp, st_loc_disp, sep=" ")
round((mt_pop_data_frac[ar_it_loc_disp, ar_it_popgrp_disp])*100, 3) %>%
  kable(caption=st_caption) %>%
  kable_styling_fc()
```

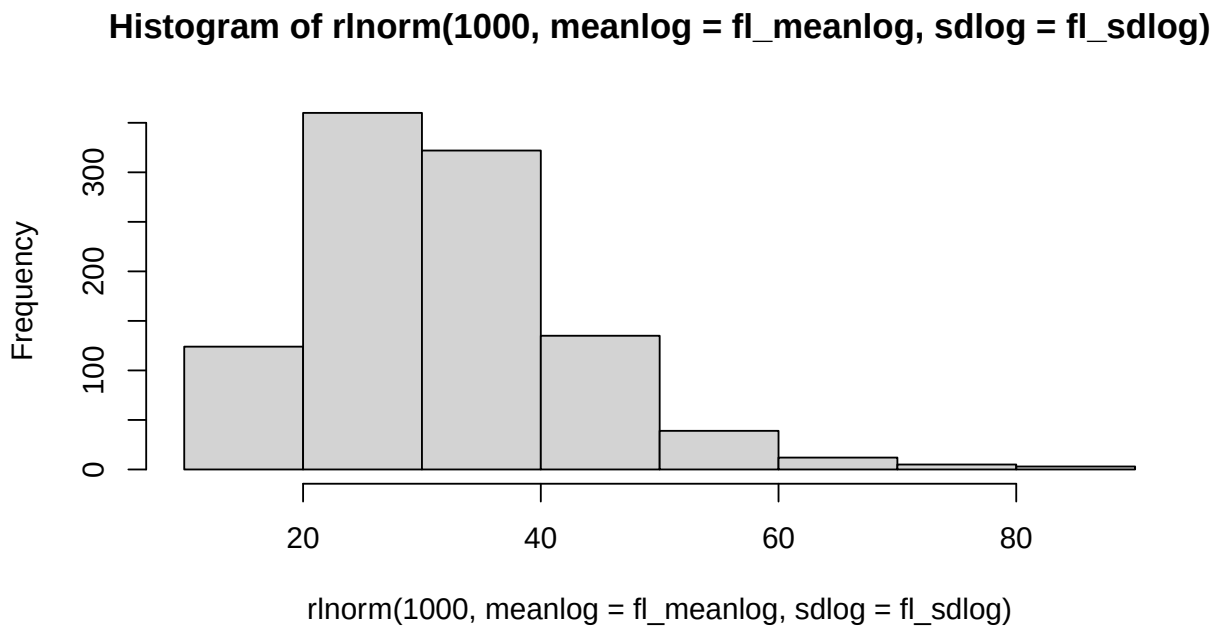
Share of population in each location and demographic cell (7 of 100 pop-groups shown) (10 of 20 locations shown)

	popgrp1	popgrp18	popgrp34	popgrp50	popgrp67	popgrp84	popgrp100
location1	0.218	0.000	0.000	0.000	0.000	0.000	0.000
location3	0.001	0.000	0.000	0.000	0.000	0.000	0.000
location5	0.000	0.009	0.122	0.000	0.000	0.000	0.000
location7	0.000	0.000	0.041	0.238	0.000	0.000	0.000
location9	0.000	0.000	0.000	0.325	0.057	0.000	0.000
location12	0.000	0.000	0.000	0.008	0.792	0.000	0.000
location14	0.000	0.000	0.000	0.000	0.194	0.059	0.000
location16	0.000	0.000	0.000	0.000	0.020	0.175	0.000
location18	0.000	0.000	0.000	0.000	0.000	0.173	0.000
location20	0.000	0.000	0.000	0.000	0.000	0.001	0.001

1.3 Simulate Environmental Exposure

Use log-normal distribution to describe average daily PM10 exposures distribution by locality:

```
fl_meanlog <- 3.4
fl_sdlog <- 0.35
hist(rlnorm(1000, meanlog = fl_meanlog, sdlog = fl_sdlog))
```



First, draw pollution measure for each locality:

```
# draw
set.seed(123)
ar_pollution_loc <- rlnorm(it_M_location, meanlog = fl_meanlog, sdlog = fl_sdlog)
# pollution dataframe
# 5 by 3 matrix
```

```r
# Column Names
ar_st_varnames <- c('location','avgdailypm10')

# Combine to tibble, add name col1, col2, etc.
tb_loc_pollution <- as_tibble(ar_pollution_loc) %>%
  rowid_to_column(var = "id") %>%
  rename_all(~c(ar_st_varnames)) %>%
  mutate(location = paste0('location', location))

# Display
st_caption = paste('PM10 Exposure across locations', st_loc_disp, sep=" ")
tb_loc_pollution[ar_it_loc_disp,] %>%
  kable(caption = st_caption) %>% kable_styling_fc()
```

PM10 Exposure across locations (10 of 20 locations shown)

location	avgdailypm10
location1	24.62676
location3	51.70466
location5	31.35114
location7	35.20967
location9	23.56121
location12	33.98553
location14	31.14765
location16	56.00380
location18	15.05461
location20	25.39426

Second, reshape population data:

```r
# Reshape population data, so each observation is location/demo
df_pop_data_frac_long <- as_tibble(mt_pop_data_frac, rownames='location') %>%
  pivot_longer(cols = starts_with('popgrp'),
               names_to = c('popgrp'),
               names_pattern = paste0("popgrp(.*)"),
               values_to = "pop_frac")
```

Third, join with pollution data:

```r
# Reshape population data, so each observation is location/demo
df_pop_pollution_long <- df_pop_data_frac_long %>%
  left_join(tb_loc_pollution, by='location')

# display
st_caption = paste('Population x Location Long Frame (15 rows shown)', sep=" ")
df_pop_pollution_long[
  round(seq(1, dim(df_pop_pollution_long)[1], length.out=15)),] %>%
  kable(caption = st_caption) %>% kable_styling_fc()
```

1.4 Compute Demographic Group Specific Exposure Distributions

What is the p10, median, p90 and mean pollution exposure for each demographic group?

1. group by population group
2. sort by pollution exposure within group

Population x Location Long Frame (15 rows shown)

location	popgrp	pop_frac	avgdailypm10
location1	1	0.0021767	24.62676
location2	44	0.0000000	27.64481
location3	87	0.0000000	51.70466
location5	29	0.0022435	31.35114
location6	72	0.0000000	54.61304
location8	15	0.0000000	19.24456
location9	58	0.0063374	23.56121
location10	100	0.0000000	25.63653
location12	43	0.0000004	33.98553
location13	86	0.0000427	34.47623
location15	29	0.0000000	24.66674
location16	72	0.0018508	56.00380
location18	14	0.0000000	15.05461
location19	57	0.0000000	38.30094
location20	100	0.0000065	25.39426

3. generate population group specific conditional population weights
4. generate population CDF for each population group (sorted by pollution)

```
# Follow four steps above
df_pop_pollution_by_popgrp_cdf <- df_pop_pollution_long %>%
  arrange(popgrp, avgdailypm10) %>%
  group_by(popgrp) %>%
  mutate(cdf_pop_condi_popgrp_sortpm10 = cumsum(pop_frac/sum(pop_frac)),
         pmf_pop_condi_popgrp_sortpm10 = (pop_frac/sum(pop_frac)))

# Display
st_caption = paste('Distribution within groups, sorted CDFs (15 rows shown)', sep=" ")
df_pop_pollution_by_popgrp_cdf[
  round(seq(1, dim(df_pop_pollution_by_popgrp_cdf)[1], length.out=15)),] %>%
  kable(caption = st_caption) %>% kable_styling_fc_wide()
```

Distribution within groups, sorted CDFs (15 rows shown)

location	popgrp	pop_frac	avgdailypm10	cdf_pop_condi_popgrp_sortpm10	pmf_pop_condi_popgrp_sortpm10
location18	1	0.0000000	15.05461	0.0000000	0.0000000
location1	15	0.0000000	24.62676	0.0000000	0.0000000
location10	21	0.0000000	25.63653	0.0000000	0.0000000
location4	28	0.0000031	30.71275	0.0006858	0.0006854
location12	34	0.0000000	33.98553	0.2666791	0.0000000
location17	40	0.0000000	35.66778	0.8019330	0.0000000
location3	47	0.0000000	51.70466	0.9971283	0.0000000
location16	53	0.0000000	56.00380	1.0000000	0.0000001
location9	60	0.0049897	23.56121	0.2860777	0.1673161
location20	67	0.0000000	25.39426	0.1045665	0.0000000
location4	73	0.0000000	30.71275	0.2102459	0.0000000
location12	8	0.0000000	33.98553	0.1625804	0.0000000
location7	86	0.0000000	35.20967	0.6805744	0.0000000
location11	92	0.0000000	45.99021	0.9988046	0.0000000
location16	99	0.0000000	56.00380	1.0000000	0.0000002

1.5 Various Relative Burden Statistics

Compute:

1. Excess pollution burden: Share of pollution burden by population group and overall population share, this is simply the ratio of population group mean and the overall weighted mean.
2. What is the fraction of the people in each population group with below and above overall average?
3. Overall percentiles, across group percentiles, and within group percentiles. Compute corresponding p80 to p20 ratios.

1.5.1 Group-specific Means and Excess pollution burden

We compute within group means, $\mathcal{Z}_{i,y}$. We compute excess population burden, $\mathcal{E}_{i,y}$, *pm10_grp_exc_burden*. 0.10 means 10 percent in excess, this means the pollution burden share is 10 percent in excess of the population share. -0.10 means 10 percent less than what population share is.

Additionally, we compute the share of people within group above the overall mean: *pm10_grp_shr_exc*. This shows the share of people having excess burden. This complements the first number. Because the 10 percent excess could be due to very high exposure to a very small number of people within a population group, or it could be that most people in the group are in "excess".

```
# Stats 1: excess pollution burden
df_excess_pollution_burden <- df_pop_pollution_by_popgrp_cdf %>%
  ungroup() %>%
  mutate(pm10_overall_mean = weighted.mean(avgdailypm10, pop_frac)) %>%
  group_by(popgrp) %>%
  mutate(
    popgrp_mass = sum(pop_frac), # The share of population for this group
    pm10_grp_mean = weighted.mean(avgdailypm10, pop_frac) # Pop-group mean
  ) %>%
  slice(1) %>%
  mutate(pm10_grp_exc_burden = pm10_grp_mean/pm10_overall_mean - 1) %>%
  select(popgrp, popgrp_mass,
         pm10_grp_mean, pm10_overall_mean, pm10_grp_exc_burden)
fl_pm10_overall_mean <- mean(df_excess_pollution_burden %>% pull(pm10_overall_mean))

# Stats 2: share of people within group below or above overall mean
df_share_below_or_excess <- df_pop_pollution_by_popgrp_cdf %>%
  arrange(popgrp, avgdailypm10) %>%
  filter(avgdailypm10 < fl_pm10_overall_mean) %>%
  slice_tail() %>%
  mutate(pm10_grp_shr_exc = 1 - cdf_pop_condi_popgrp_sortpm10) %>%
  select(popgrp, pm10_grp_shr_exc)
# merge stats 2 with stats 1
df_excess_pollution_burden <- df_excess_pollution_burden %>%
  left_join(df_share_below_or_excess, by="popgrp")

# display
st_caption = paste('Mean and Excess Burden by Population Groups',
  st_popgrp_disp, sep=" ")
df_excess_pollution_burden[ar_it_popgrp_disp,] %>%
  kable(caption = st_caption) %>%
  kable_styling_fc_wide()
```

Mean and Excess Burden by Population Groups (7 of 100 pop-groups shown)

popgrp	popgrp_mass	pm10_grp_mean	pm10_overall_mean	pm10_grp_exc_burden	pm10_grp_shr_exc
1	0.0028472	25.41850	33.25198	-0.2355794	0.0033669
24	0.0020059	39.83004	33.25198	0.1978245	0.3655482
39	0.0035380	40.43425	33.25198	0.2159951	0.9256668
53	0.0203775	28.38652	33.25198	-0.1463210	0.2803180
69	0.0218601	33.18168	33.25198	-0.0021141	0.6637586
84	0.0064339	34.11731	33.25198	0.0260235	0.5404550
99	0.0001213	33.03851	33.25198	-0.0064196	0.5953337

1.5.2 Within Group Percentiles

Compute within group percentiles for each population groups. Use the list of percentiles below to specify which percentiles should be computed.

```
# Stats 3: percentiles and ratios
# Stats 3a: generate key within group percentiles
# 1. 20th and 80th percentiles
# 2. 10th and 90th percentiles
# 3. 50th percentile
# Generate pollution quantiles by population groups
for (it_percentile_ctr in seq(1, length(ar_fl_percentiles))) {

  # Current within group percentile to compute
  fl_percentile <- ar_fl_percentiles[it_percentile_ctr]
  svr_percentile <- paste0('pm10_p', round(fl_percentile*100))

  # Frame with specific percentile
  df_within_percentiles_cur <- df_pop_pollution_by_popgrp_cdf %>%
    group_by(popgrp) %>%
    filter(cdf_pop_condi_popgrp_sortpm10 >= fl_percentile) %>%
    slice(1) %>%
    mutate(!!sym(svr_percentile) := avgdailypm10) %>%
    select(popgrp, one_of(svr_percentile))

  # Merge percentile frames together
  if (it_percentile_ctr > 1) {
    df_within_percentiles <- df_within_percentiles %>%
      left_join(df_within_percentiles_cur, by='popgrp')
  } else {
    df_within_percentiles <- df_within_percentiles_cur
  }
}

# display
st_caption = paste('PM10 Exposure Distribution by Population Groups',
  st_popgrp_disp, sep=" ")
df_within_percentiles[ar_it_popgrp_disp,] %>%
  kable(caption = st_caption) %>%
  kable_styling_fc()
```

PM10 Exposure Distribution by Population Groups (7 of 100 pop-groups shown)

popgrp	pm10_p10	pm10_p20	pm10_p80	pm10_p90
1	24.62676	24.62676	27.64481	27.64481
24	31.35114	31.35114	54.61304	54.61304
39	35.20967	35.20967	54.61304	54.61304
53	19.24456	19.24456	45.99021	45.99021
69	24.66674	31.14765	34.47623	34.47623
84	15.05461	15.05461	56.00380	56.00380
99	25.39426	25.39426	38.30094	38.30094

1.5.3 Overall and Across Group Percentiles

First, use the *df_pop_pollution_by_popgrp_cdf* dataframe, and get total population mass by location, and average polllution per location (raw data). Note that *avgdailypm10* is unifom within location.

```
df_location_mean <- df_pop_pollution_by_popgrp_cdf %>%
  ungroup() %>%
  group_by(location) %>%
  mutate(
    location_mass = sum(pop_frac), # The share of population for this group
    pm10_mean = weighted.mean(avgdailypm10, pop_frac) # Pop-group mean, don't need this, common within
  ) %>%
  slice(1) %>%
  ungroup() %>%
  arrange(pm10_mean) %>%
  mutate(cdf_sortpm10 = cumsum(location_mass)) %>%
  select(location, location_mass, cdf_sortpm10, pm10_mean) %>%
  mutate(popgrp = "overall")

# display
st_caption = paste('PM10 Exposure Distribution Overall', st_loc_disp, sep=" ")
df_location_mean[ar_it_loc_disp,] %>%
  kable(caption = st_caption) %>%
  kable_styling_fc_wide()
```

Second, use the just created *df_excess_pollution_burden* dataframe, and get population by population groups and mean exposures.

```
df_popgrp_mean <- df_excess_pollution_burden %>%
  ungroup() %>%
  arrange(pm10_grp_mean) %>%
  mutate(cdf_sortpm10 = cumsum(popgrp_mass)) %>%
  select(popgrp, popgrp_mass, cdf_sortpm10, pm10_grp_mean) %>%
  rename(pm10_mean = pm10_grp_mean) %>%
  mutate(popgrp = "across-group")

# display
st_caption = paste('PM10 Exposure Across Groups', st_popgrp_disp, sep=" ")
df_popgrp_mean[ar_it_popgrp_disp,] %>%
  kable(caption = st_caption) %>%
  kable_styling_fc_wide()
```

Third, use the same percentile calculation structure as earlier and compute overall and across group percentiles.

PM10 Exposure Distribution Overall (10 of 20 locations shown)

location	location_mass	cdf_sortpm10	pm10_mean	popgrp
location18	0.0252963	0.0252963	15.05461	overall
location9	0.0849047	0.1796685	23.56121	overall
location15	0.0694675	0.2515870	24.66674	overall
location10	0.0970339	0.3577247	25.63653	overall
location4	0.0157247	0.3783515	30.71275	overall
location12	0.1037259	0.5922784	33.98553	overall
location7	0.0531222	0.7424345	35.20967	overall
location19	0.0157247	0.7961037	38.30094	overall
location3	0.0091038	0.9089334	51.70466	overall
location16	0.0531222	1.0000000	56.00380	overall

PM10 Exposure Across Groups (7 of 100 pop-groups shown)

popgrp	popgrp_mass	cdf_sortpm10	pm10_mean
across-group	0.0041575	0.0041575	21.30080
across-group	0.0281559	0.1964059	29.64248
across-group	0.0009975	0.3556931	31.99898
across-group	0.0209968	0.5655672	33.28429
across-group	0.0233032	0.8423269	36.37059
across-group	0.0039372	0.9423163	42.45066
across-group	0.0016343	1.0000000	49.36976

```
for (it_df in c(1,2)) {
  for (it_percentile_ctr in seq(1, length(ar_fl_percentiles))) {

    # Load in data-frames
    if (it_df == 1) {
      df_working_inputs <- df_location_mean
    } else if (it_df == 2) {
      df_working_inputs <- df_popgrp_mean
    }

    # Current within group percentile to compute
    fl_percentile <- ar_fl_percentiles[it_percentile_ctr]
    svr_percentile <- paste0('pm10_p', round(fl_percentile*100))

    # Frame with specific percentile
    df_within_percentiles_cur <- df_working_inputs %>%
```

```
      filter(cdf_sortpm10 >= fl_percentile) %>%
      slice(1) %>%
      mutate(!!sym(svr_percentile) := pm10_mean) %>%
      select(popgrp, one_of(svr_percentile))

    # Merge percentile frames together
    if (it_percentile_ctr > 1) {
      df_percentiles <- df_percentiles %>%
        left_join(df_within_percentiles_cur, by='popgrp')
    } else {
      df_percentiles <- df_within_percentiles_cur
    }
  }

  if (it_df == 1) {
    df_location_mean_perc <- df_percentiles
  } else if (it_df == 2) {
    df_popgrp_mean_perc <- df_percentiles
  }

}

# Stack results together

# display
st_caption = paste('Overall PM10 Distribution')
df_location_mean_perc %>%
  kable(caption = st_caption) %>%
  kable_styling_fc()
```

Overall PM10 Distribution

popgrp	pm10_p10	pm10_p20	pm10_p80	pm10_p90
overall	23.56121	24.66674	45.99021	51.70466

```
st_caption = paste('Across Population Group PM10 Distribution')
df_popgrp_mean_perc %>%
  kable(caption = st_caption) %>%
  kable_styling_fc()
```

Across Population Group PM10 Distribution

popgrp	pm10_p10	pm10_p20	pm10_p80	pm10_p90
across-group	28.67913	30.13912	35.65987	37.78554

1.5.4 Combine All Percentiles

Combine percentiles within, overall and across groups, together with excess burden results.

```
# Percentiles and excess burden data combined
df_excburden_percentiles <- df_excess_pollution_burden %>%
  left_join(df_within_percentiles, by="popgrp")
# Overall and Across Group percentiles
df_excburden_percentiles <- bind_rows(df_excburden_percentiles, df_location_mean_perc, df_popgrp_mean_p
```

1.5.5 P80 to P20 Relative Exposures

We now compute the relative percentile ratios of interest. Given that we have merged our dataframes, we include here within group, across group, and overall percentile ratio results.

```
# lower and upper bound or relative within group ratios
# can only use values appearing in the percentiles list prior
# Stats 4c: Ratios
# Generate P80 to P20 ratio, and P90 to P10 standard inequality ratios
for (it_ratio_ctr in seq(1, length(ar_fl_ratio_upper))) {

  # Upper and lower percentile bounds
  fl_ratio_upper <- ar_fl_ratio_upper[it_ratio_ctr]
  fl_ratio_lower <- ar_fl_ratio_lower[it_ratio_ctr]
  svr_ratio_upper_perc <- paste0('pm10_p', round(fl_ratio_upper*100))
  svr_ratio_lower_perc <- paste0('pm10_p', round(fl_ratio_lower*100))

  # New relative within group ratio variable name
  svr_ratio <- paste0('pm10_rat_p', round(fl_ratio_upper*100), '_dvd_p', round(fl_ratio_lower*100))

  # Generate P80 to P20 ratio, etc.
  df_excburden_percentiles  <- df_excburden_percentiles  %>%
    mutate(!!sym(svr_ratio) := !!sym(svr_ratio_upper_perc)/!!sym(svr_ratio_lower_perc))
}

# display
st_caption = paste('PM10 Exposure within/across/overall Population Group P80-P20 Inequality',
  st_popgrp_disp, sep=" ")
df_excburden_percentiles[ar_it_popgrp_disp_withoverall,] %>%
  select(-pm10_overall_mean,
         -starts_with('pm10_grp_excbrd_p'),
         -starts_with('pm10_p')) %>%
  kable(caption = st_caption) %>%
  kable_styling_fc_wide()
```

PM10 Exposure within/across/overall Population Group P80-P20 Inequality (7 of 100 pop-groups shown)

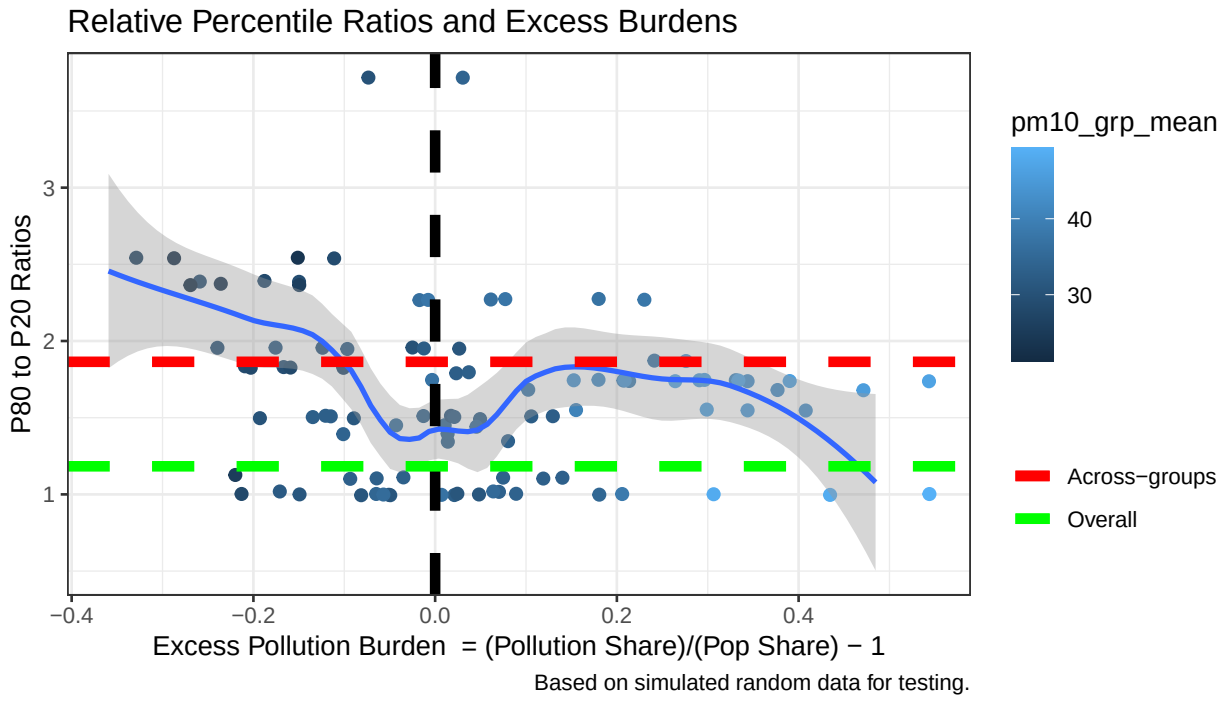
popgrp	popgrp_mass	pm10_grp_mean	pm10_grp_exc_burden	pm10_grp_shr_exc	pm10_rat_p80_dvd_p20	pm10_rat_p90_dvd_p10
1	0.0028472	25.41850	-0.2355794	0.0033669	1.122552	1.122552
24	0.0020059	39.83004	0.1978245	0.3655482	1.741979	1.741979
39	0.0035380	40.43425	0.2159951	0.9256668	1.551081	1.551081
53	0.0203775	28.38652	-0.1463210	0.2803180	2.389777	2.389777
69	0.0218601	33.18168	-0.0021141	0.6637586	1.106864	1.397681
84	0.0064339	34.11731	0.0260235	0.5404550	3.720044	3.720044
99	0.0001213	33.03851	-0.0064196	0.5953337	1.508252	1.508252
overall	NA	NA	NA	NA	1.864463	2.194483
across-group	NA	NA	NA	NA	1.183175	1.317527

Visualize the distribution of within group relative percentiles

```
# Rounding excess burden s.d.
df_scatter_main <- df_excburden_percentiles %>%
  filter(!popgrp %in% c("overall", "across-group"))
fl_pm10_rat_p80_dvd_p20_overall <- mean(df_excburden_percentiles %>%
  filter(popgrp %in% c("overall")) %>% pull(pm10_rat_p80_dvd_p20))
fl_pm10_rat_p80_dvd_p20_acrossgrp <- mean(df_excburden_percentiles %>%
  filter(popgrp %in% c("across-group")) %>% pull(pm10_rat_p80_dvd_p20))
st_title <- paste0("Relative Percentile Ratios and Excess Burdens")
# title_line1 <- paste0("Histogram shows the distribution of Relative Ratios")
```

```r
# Generate a Data Sample by Drawing from the Distribution
it_sample_draws <- 1e6
# ar_it_draws <- sample(1:it_N_pop_groups, it_sample_draws, replace=TRUE, prob=ar_data_grp_shares)
# ar_sample_draws <- ar_data_grp_exc_burden[ar_it_draws]
# Draw histogram
pl_excess_burden <- df_scatter_main %>%
  ggplot(aes(x=pm10_grp_exc_burden,
             y=pm10_rat_p80_dvd_p20)) +
  geom_jitter(aes(size=popgrp_mass, color=pm10_grp_mean), width = 0.15, size=2) +
  geom_smooth(span = 0.50, se=TRUE) +
  theme_bw() +
  geom_vline(aes(xintercept=0), color="black", linetype="dashed", size=2) +
  geom_hline(aes(yintercept=fl_pm10_rat_p80_dvd_p20_overall,
                 linetype="Overall"),
             color="red", size=2) +
  geom_hline(aes(yintercept=fl_pm10_rat_p80_dvd_p20_acrossgrp,
                 linetype="Across-groups"),
             color="green", size=2) +
  labs(title = st_title,
       # subtitle = paste0(title_line1),
       x = 'Excess Pollution Burden  = (Pollution Share)/(Pop Share) - 1',
       y = 'P80 to P20 Ratios',
       caption = 'Based on simulated random data for testing.') +
  scale_linetype_manual(name = "", values = c(2, 2),
                        guide = guide_legend(override.aes = list(color = c("red", "green"))))

# Print
print(pl_excess_burden)
```



1.6 Across Group Atkinson And Gini Statistics

Whether we are looking at inequality within group, or inequality across groups, we can compute various aggregate inequality related statistics, including:

1. Gini
2. Atkinson
3. Standard deviation
4. P80 to P20 ratio

The literature has focused on Atkinson statistics computed over mean pollution exposures across groups. When we focus on variations across groups, we compute these based on group means. When we focus on variations within groups, we compute group-specific GINI, Atkinson, etc based on within group data.

Earlier, we have already computed the P80 to P20 ratio for each population group.

1.6.1 Inequality in Group Means

We compute a number of aggregate statistics over the means of the subgroups. We can only compute 1 gini and 1 standard deviation, but we have a spectrum of Atkinson index values, depending in the inequality aversion value picked.

First, we get the data inputs we need.

```
# 1. SORT FIRST!
df_excess_pollution_burden_sorted <- df_excess_pollution_burden %>% arrange(pm10_grp_mean)
# 2. Obtain the means across groups, and also excess burden across groups
ar_data_grp_means <- df_excess_pollution_burden_sorted %>% pull(pm10_grp_mean)
ar_data_grp_exc_burden <- df_excess_pollution_burden_sorted %>% pull(pm10_grp_exc_burden)
# 3. Obtain the probability mass for each group
ar_data_grp_shares <- df_excess_pollution_burden_sorted %>% pull(popgrp_mass)
```

Second, we compute the GINI coefficient, standard deviation, and coefficient of variations over group-specific means, with group-specific population shares.

```
# compute gini over group means, and standard deviations
fl_grp_means_gini <- ffi_dist_gini_random_var_pos_test(ar_data_grp_means, ar_data_grp_shares)
# STD, and coefficient of variation
ls_fl_grp_means_std_cov <- ffi_std_cov(ar_data_grp_means, ar_data_grp_shares)
fl_grp_means_std <- ls_fl_grp_means_std_cov$std
fl_grp_means_cov <- ls_fl_grp_means_std_cov$cov
```

Third, compute an array of Atkinson index with differing inequality aversions.

```
# Log10 scaled Inequality Measures
ar_rho <- 1 - (10^(c(seq(-2,2, length.out=30))))
tb_rho <- as_tibble(unique(ar_rho))
# Array of atkinson values
ar_grp_means_atkinson <- apply(tb_rho, 1, function(row){
  ffi_atkinson_random_var_ineq(ar_data_grp_means, ar_data_grp_shares, row[1])})
# atkinson results table
ar_st_varnames <- c('rho_id','rho','atkinson_index')
tb_atkinson <- as_tibble(cbind(ar_rho, ar_grp_means_atkinson)) %>%
  rowid_to_column(var = "id") %>%
  rename_all(~c(ar_st_varnames)) %>%
  mutate(one_minus_rho = 1 - rho) %>%
  select(rho_id, one_minus_rho, rho, atkinson_index)
# Max Atkinson
fl_atkinson_max <- max(tb_atkinson %>% pull(atkinson_index))
```

```
# display
it_rows_shown <- 10
st_caption <- paste0('Atkinson Inequality Index',
  '(', it_rows_shown, ' of ', length(ar_rho), ' inequality preferences shown)')
tb_atkinson[round(seq(1, length(ar_rho), length.out = it_rows_shown)),] %>%
  kable(caption = st_caption) %>%
  kable_styling_fc()
```

Atkinson Inequality Index(10 of 30 inequality preferences shown)

rho_id	one_minus_rho	rho	atkinson_index
1	0.0100000	0.9900000	0.0000826
4	0.0259294	0.9740706	0.0002141
7	0.0672336	0.9327664	0.0005545
11	0.2395027	0.7604973	0.0019672
14	0.6210169	0.3789831	0.0050568
17	1.6102620	-0.6102620	0.0128582
20	4.1753189	-3.1753189	0.0324195
24	14.8735211	-13.8735211	0.1336964
27	38.5662042	-37.5662042	0.2671924
30	100.0000000	-99.0000000	0.3236560

Fourth, we now create a row that shows three selected Atkinson, Gini, as well as the coefficient of variation.

```
# Selected five atkinson
ar_it_rho_select <- round(seq(1, length(ar_rho), length.out=5))
ar_atk_selected <- ar_grp_means_atkinson[ar_it_rho_select]
ar_rho_selected <- round(ar_rho[ar_it_rho_select], 2)
ar_st_rho_selected_colnames <- paste0("atk4rho=", ar_rho_selected)

# all stats together
mt_cov_gini_atk <- matrix(
  c(fl_grp_means_cov, fl_grp_means_gini, ar_atk_selected),
    nrow=1, ncol=(2+length(ar_it_rho_select)))

# Display results as table
ar_st_varnames <- c('cov', 'gini', ar_st_rho_selected_colnames)
tb_cov_gini_atk <- as_tibble(mt_cov_gini_atk) %>%
  rename_all(~c(ar_st_varnames))

# display
st_caption <- paste0("Coefficient of Variation (COV), Gini,",
                     "and three selected Atkinson Measures",
                     "computed given group means, with different weights",
                     "(share of population) for each group.")
tb_cov_gini_atk %>%
  kable(caption = st_caption) %>%
  kable_styling_fc_wide()
```

Fifth, we can compute the standard deviation of excess burden. Note that excess burden is both positive and negative, so we can not compute GINI or Atkinson statistics from it. Note that by construction the standard deviation of excess burden is the same as the coefficient of variation for the underlying raw means.

Coefficient of Variation (COV), Gini,and three selected Atkinson Measurescomputed given group means, with different weights(share of population) for each group.

cov	gini	atk4rho=0.99	atk4rho=0.91	atk4rho=-0.17	atk4rho=-9.83	atk4rho=-99
0.1305074	0.0667071	8.26e-05	0.0007613	0.0094356	0.0906065	0.323656

```
ls_fl_std_cov_grp_exc_burden <- ffi_std_cov(ar_data_grp_exc_burden, ar_data_grp_shares)
fl_grp_exc_burden_std <- ls_fl_std_cov_grp_exc_burden$std
st_disp <- paste("Coefficient of Variation (COV) of raw means =", fl_grp_means_cov,
                 "is always the same as the Standard deviation of excess burden =",
                 fl_grp_exc_burden_std)
print(st_disp)
```

```
## [1] "Coefficient of Variation (COV) of raw means = 0.130507412312003 is always the same as the Standa
```

1.6.2 Visualize Inequality in Group Means (GINI and ATKINSON)

Now we can present what the existing literature focuses on, which is to compute Atkinson index over group means. As can be seen in the figure below, this index value is arbitrary depending on inequality aversion. Additionally, this value shows in this testing example, very low inequality in pollution exposure. Additionally, we also have the GINI index, which can be thought of as mapping to a particular level of inequality aversion.

Potentially, an issue with using these types of measures is that they are hard to interpret, potentially arbitrary, and potentially generates misleading impressions. In the test examples here, there is almost equality using GINI and Atkinson measures.

Our excess pollution burden statistics, however, provides a more clearly interpretable lense for looking at across group inequality. Additionally, we also look at within group inequality.

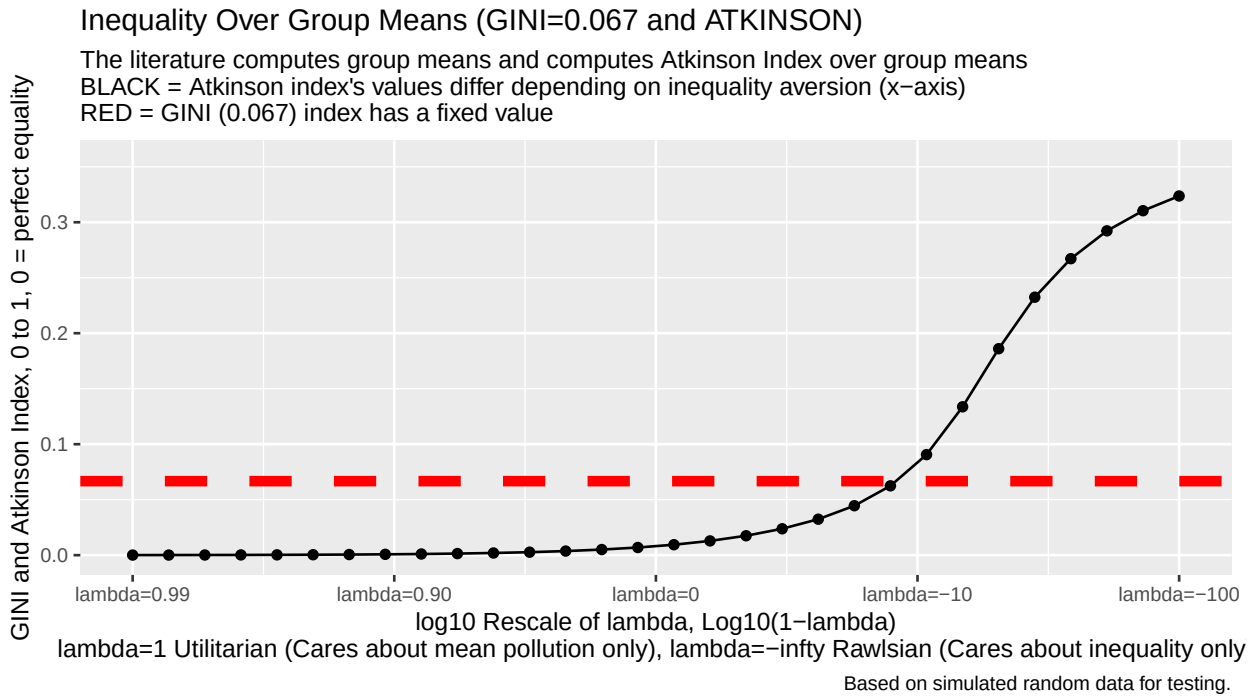
```
# x-labels
x.labels <- c('lambda=0.99', 'lambda=0.90', 'lambda=0', 'lambda=-10', 'lambda=-100')
x.breaks <- c(0.01, 0.10, 1, 10, 100)

# title line 2
fl_grp_means_gini_fmt <- round(fl_grp_means_gini, 3)
st_title <- paste0("Inequality Over Group Means (GINI=", fl_grp_means_gini_fmt," and ATKINSON)")
title_line1 <- paste0("The literature computes group means and computes Atkinson Index over group means
title_line2 <- paste0("BLACK = Atkinson index's values differ depending on inequality aversion (x-axis)
title_line3 <- paste0("RED = GINI (", fl_grp_means_gini_fmt,") index has a fixed value")

# Graph Results--Draw
pl_gini_atkinson <- tb_atkinson %>%
  ggplot(aes(x=one_minus_rho, y=atkinson_index)) +
  geom_line() +
  geom_point() +
  geom_hline(yintercept=fl_grp_means_gini, linetype='dashed', color='red', size=2) +
  # geom_vline(xintercept=c(1), linetype="dotted") +
  labs(title = st_title,
       subtitle = paste0(title_line1, '\n', title_line2, '\n', title_line3),
       x = 'log10 Rescale of lambda, Log10(1-lambda)\nlambda=1 Utilitarian (Cares about mean pollution
       y = paste0('GINI and Atkinson Index, 0 to 1, 0 = perfect equality'),
       caption = 'Based on simulated random data for testing.') +
  scale_x_continuous(trans='log10', labels = x.labels, breaks = x.breaks) +
  ylim(0, max(min(fl_atkinson_max*1.1, 1), fl_grp_means_gini)) +
  theme(text = element_text(size = 10),
```

```
        legend.position="right")

# Print
print(pl_gini_atkinson)
```



1.6.3 Visualize Inequality in Group Means (Excess Burden)

Now we visualize excess pollution burden as a distribution, this is our preferred statistics to look at variations in means across groups.

```
# Rounding excess burden s.d.
fl_grp_exc_burden_std_fmt <- round(fl_grp_exc_burden_std, 3)
st_title <- paste0("Distribution of Excess Burden (Across Group Variation), s.d.=", fl_grp_exc_burden_s
title_line1 <- paste0("Histogram shows the distribution of excess burden by population groups")
title_line2 <- paste0("Excess Burden = (Pollution Share)/(Pop Share) - 1")

# Generate a Data Sample by Drawing from the Distribution
it_sample_draws <- 1e6
ar_it_draws <- sample(1:it_N_pop_groups, it_sample_draws, replace=TRUE, prob=ar_data_grp_shares)
ar_sample_draws <- ar_data_grp_exc_burden[ar_it_draws]
# Draw histogram
pl_excess_burden <- as_tibble(ar_sample_draws) %>%
  ggplot(aes(x=value)) +
  # geom_histogram(aes(y=..density..),
  #                colour="darkblue", fill="lightblue")+
  geom_density(alpha=.2, fill="#FF6666") +
  geom_vline(aes(xintercept=0),
             color="blue", linetype="dashed", size=2) +
  labs(title = st_title,
       subtitle = paste0(title_line1, '\n', title_line2),
```

```
       x = 'Excess Pollution Burden  = (Pollution Share)/(Pop Share) - 1',
       y = 'Density',
       caption = 'Based on simulated random data for testing.')
# Print
print(pl_excess_burden)
```