

# Inequality in Environmental Exposure Across Population Groups

Fan Wang

2021-02-02

## Contents

<b>1</b>	<b>Location, Population, and Pollution</b>	<b>1</b>
1.1	Simulate Population Distribution over Location and Demographics . . . . .	1
1.2	Simulate Enviromental Exposure . . . . .	3
1.3	Compute Demographic Group Specific Exposure Distributions . . . . .	4
1.4	Compute the Gini Index by Population Subgroup . . . . .	5
1.5	Visualize the Distributions . . . . .	6
1.6	Various Quantiles . . . . .	8

## 1 Location, Population, and Pollution

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools](#) Package, [R Code Examples](#) Repository ([bookdown site](#)), or [Intro Stats with R](#) Repository ([bookdown site](#)).

### 1.1 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:

```
# 7 different age groups and 12 different locations
it_N_pop_groups <- 7
it_M_location <- 12
# 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))
# Display
mt_pop_data_frac %>% kable() %>% kable_styling_fc()
```

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)
```

	popgrp1	popgrp2	popgrp3	popgrp4	popgrp5	popgrp6	popgrp7
location1	NA	NA	NA	NA	NA	NA	NA
location2	NA	NA	NA	NA	NA	NA	NA
location3	NA	NA	NA	NA	NA	NA	NA
location4	NA	NA	NA	NA	NA	NA	NA
location5	NA	NA	NA	NA	NA	NA	NA
location6	NA	NA	NA	NA	NA	NA	NA
location7	NA	NA	NA	NA	NA	NA	NA
location8	NA	NA	NA	NA	NA	NA	NA
location9	NA	NA	NA	NA	NA	NA	NA
location10	NA	NA	NA	NA	NA	NA	NA
location11	NA	NA	NA	NA	NA	NA	NA
location12	NA	NA	NA	NA	NA	NA	NA

```

# 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 = 0.962953679726938"

# Display
round(mt_pop_data_frac*100, 2) %>%
  kable(caption='Share of population in each location and demographic cell') %>%
  kable_styling_fc()

```

Share of population in each location and demographic cell

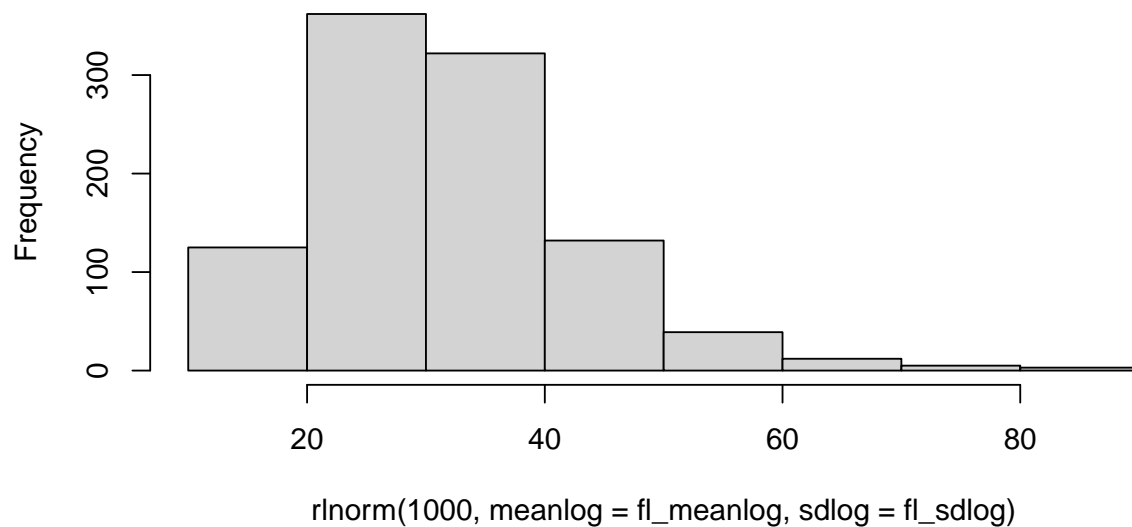
	popgrp1	popgrp2	popgrp3	popgrp4	popgrp5	popgrp6	popgrp7
location1	1.09	0.13	0.01	0.00	0.00	0.00	0.00
location2	1.63	0.70	0.13	0.01	0.00	0.00	0.00
location3	0.59	1.40	1.39	0.74	0.22	0.03	0.00
location4	0.06	0.43	1.29	2.09	1.90	0.92	0.19
location5	0.07	0.55	1.73	2.89	2.71	1.36	0.28
location6	0.02	0.26	1.19	2.89	3.93	2.85	0.86
location7	0.01	0.10	0.66	2.23	4.26	4.33	1.83
location8	0.00	0.06	0.47	1.83	4.03	4.72	2.31
location9	0.00	0.03	0.27	1.26	3.28	4.55	2.63
location10	0.00	0.02	0.20	0.96	2.57	3.68	2.19
location11	0.00	0.00	0.00	0.04	0.40	2.05	4.38
location12	0.00	0.00	0.00	0.02	0.24	1.28	2.82

## 1.2 Simulate Enviromental 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))
```

**Histogram of `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

# 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
kable(tb_loc_pollution) %>% kable_styling_fc()
```

Second, reshape population data:

```
# Reshape population data, so each observation is location/demo
df_pop_data_frac_long <- as_tibble(mt_pop_data_frac, rownames='location') %>%
```

location	avgdailypm10
location1	24.62676
location2	27.64481
location3	51.70466
location4	30.71275
location5	31.35114
location6	54.61304
location7	35.20967
location8	19.24456
location9	23.56121
location10	25.63653
location11	45.99021
location12	33.98553

```

pivot_longer(cols = starts_with('popgrp'),
  names_to = c('popgrp'),
  names_pattern = paste0("popgrp(.*)"),
  values_to = "pop_frac")

```

Third, join with pollution data:

```

# 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
df_pop_pollution_long[1:round(it_N_pop_groups*2.5),] %>% kable() %>% kable_styling_fc()

```

location	popgrp	pop_frac	avgdailypm10
location1	1	0.0109366	24.62676
location1	2	0.0013417	24.62676
location1	3	0.0000686	24.62676
location1	4	0.0000019	24.62676
location1	5	0.0000000	24.62676
location1	6	0.0000000	24.62676
location1	7	0.0000000	24.62676
location2	1	0.0163003	27.64481
location2	2	0.0070132	27.64481
location2	3	0.0012573	27.64481
location2	4	0.0001202	27.64481
location2	5	0.0000065	27.64481
location2	6	0.0000002	27.64481
location2	7	0.0000000	27.64481
location3	1	0.0058760	51.70466
location3	2	0.0140000	51.70466
location3	3	0.0138984	51.70466
location3	4	0.0073587	51.70466

### 1.3 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
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
df_pop_pollution_by_popgrp_cdf[1:round(it_N_pop_groups*5.5),] %>%
  kable() %>% kable_styling_fc_wide()
```

location	popgrp	pop_frac	avgdailypm10	cdf_pop_condi_popgrp_sortpm10	pmf_pop_condi_popgrp_sortpm10
location8	1	0.0000364	19.24456	0.0010453	0.0010453
location9	1	0.0000151	23.56121	0.0014804	0.0004351
location1	1	0.0109366	24.62676	0.3156484	0.3141680
location10	1	0.0000104	25.63653	0.3159471	0.0002988
location2	1	0.0163003	27.64481	0.7841942	0.4682471
location4	1	0.0005879	30.71275	0.8010816	0.0168874
location5	1	0.0007392	31.35114	0.8223166	0.0212350
location12	1	0.0000000	33.98553	0.8223168	0.0000002
location7	1	0.0000681	35.20967	0.8242718	0.0019550
location11	1	0.0000000	45.99021	0.8242721	0.0000003
location3	1	0.0058760	51.70466	0.9930669	0.1687948
location6	1	0.0002413	54.61304	1.0000000	0.0069331
location8	2	0.0006400	19.24456	0.0172871	0.0172871
location9	2	0.0003150	23.56121	0.0257947	0.0085076
location1	2	0.0013417	24.62676	0.0620374	0.0362427
location10	2	0.0002235	25.63653	0.0680736	0.0060362
location2	2	0.0070132	27.64481	0.2575157	0.1894421
location4	2	0.0042712	30.71275	0.3728918	0.1153760
location5	2	0.0055479	31.35114	0.5227547	0.1498629
location12	2	0.0000004	33.98553	0.5227662	0.0000116
location7	2	0.0010378	35.20967	0.5508009	0.0280347
location11	2	0.0000008	45.99021	0.5508213	0.0000203
location3	2	0.0140000	51.70466	0.9289930	0.3781718
location6	2	0.0026287	54.61304	1.0000000	0.0710070
location8	3	0.0046896	19.24456	0.0638166	0.0638166
location9	3	0.0027290	23.56121	0.1009539	0.0371373
location1	3	0.0000686	24.62676	0.1018872	0.0009333
location10	3	0.0020006	25.63653	0.1291118	0.0272246
location2	3	0.0012573	27.64481	0.1462207	0.0171089
location4	3	0.0129304	30.71275	0.3221799	0.1759592
location5	3	0.0173492	31.35114	0.5582709	0.2360910
location12	3	0.0000141	33.98553	0.5584625	0.0001916
location7	3	0.0065945	35.20967	0.6482016	0.0897391
location11	3	0.0000242	45.99021	0.6485305	0.0003290
location3	3	0.0138984	51.70466	0.8376617	0.1891312
location6	3	0.0119295	54.61304	1.0000000	0.1623383
location8	4	0.0183277	19.24456	0.1224562	0.1224562
location9	4	0.0126118	23.56121	0.2067219	0.0842656

## 1.4 Compute the Gini Index by Population Subgroup

The Gini index from `fs_gini_disc`.

```
ffi_dist_gini_random_var_pos_test <- function(ar_x_sorted, ar_prob_of_x) {
  fl_mean <- sum(ar_x_sorted*ar_prob_of_x);
```

```

ar_mean_cumsum <- cumsum(ar_x_sorted*ar_prob_of_x);
ar_height <- ar_mean_cumsum/fl_mean;
fl_area_drm <- sum(ar_prob_of_x*ar_height);
fl_area_below45 <- sum(ar_prob_of_x*(cumsum(ar_prob_of_x)/sum(ar_prob_of_x)))
fl_gini_index <- (fl_area_below45-fl_area_drm)/fl_area_below45
return(fl_gini_index)
}

```

Compute Gini index for sub-group:

```

# Compute GINI by group
df_pop_pollu_gini <- df_pop_pollution_by_popgrp_cdf %>%
  group_by(popgrp) %>%
  do(popgrp_gini = ffi_dist_gini_random_var_pos_test(
    . $avgdailympm10, . $pmf_pop_condi_popgrp_sortpm10)) %>%
  unnest(c(popgrp_gini)) %>%
  left_join(df_pop_pollution_by_popgrp_cdf %>%
    group_by(popgrp) %>% slice(1L) %>%
    select(popgrp)
    , by="popgrp")
# Display
df_pop_pollu_gini %>% kable() %>% kable_styling_fc()

```

popgrp	popgrp_gini
1	0.1206861
2	0.1075096
3	0.1380319
4	0.1543002
5	0.1680265
6	0.1754309
7	0.1453136

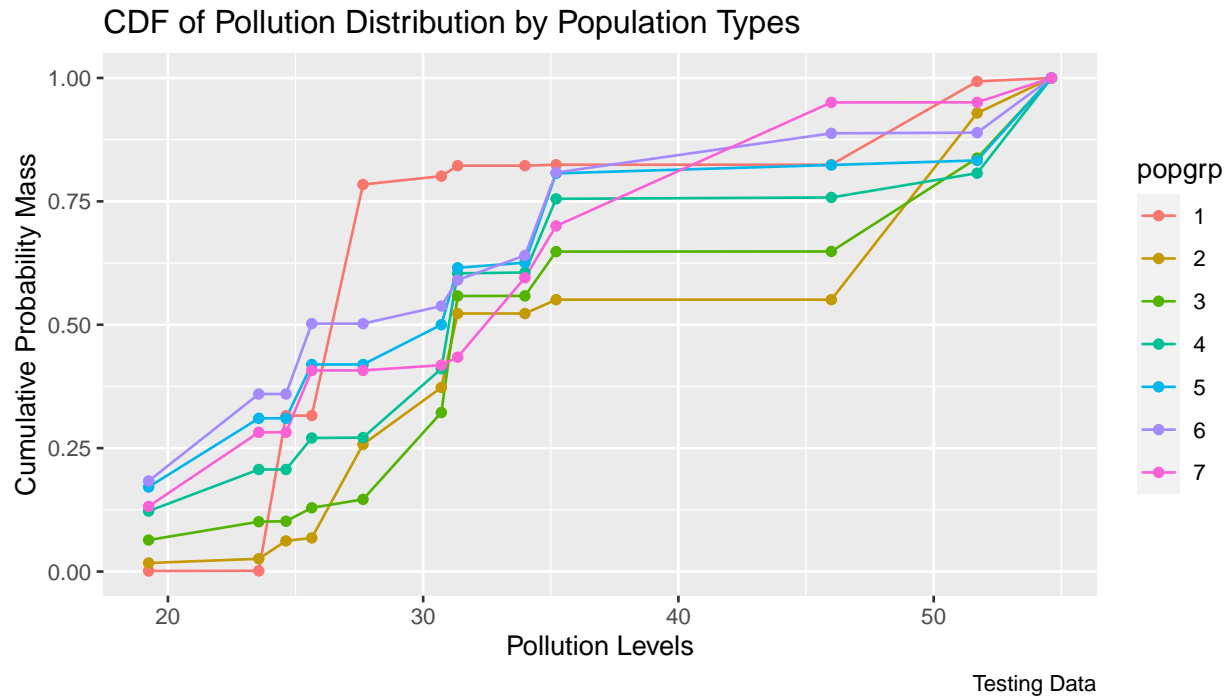
## 1.5 Visualize the Distributions

Visualizing distributions. Visualize the CDF:

```

# Visualize Distributions, CDF for different population groups
lineplot <- df_pop_pollution_by_popgrp_cdf %>%
  select(popgrp, avgdailympm10, cdf_pop_condi_popgrp_sortpm10 ) %>%
  ggplot(aes(x=avgdailympm10, y=cdf_pop_condi_popgrp_sortpm10,
    colour=popgrp)) +
  geom_line() +
  geom_point() +
  labs(title = 'CDF of Pollution Distribution by Population Types',
    x = 'Pollution Levels',
    y = 'Cumulative Probability Mass',
    caption = 'Testing Data')
print(lineplot)

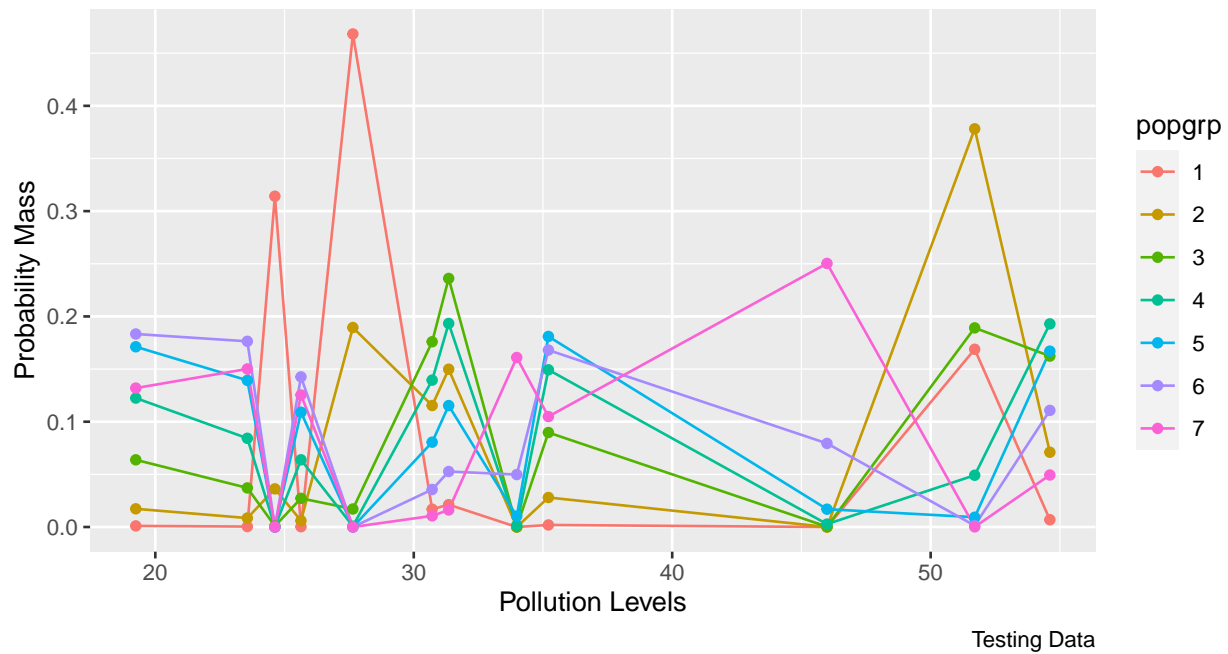
```



Visualize the Probability Mass (real data should look much less chaotic than this):

```
# Visualize Distributions, CDF for different population groups
lineplot_pmf <- df_pop_pollution_by_popgrp_cdf %>%
  select(popgrp, avgdailypm10, pmf_pop_condi_popgrp_sortpm10 ) %>%
  ggplot(aes(x=avgdailypm10, y=pmf_pop_condi_popgrp_sortpm10,
             colour=popgrp)) +
    geom_line() +
    geom_point() +
    labs(title = 'Prob Mass Func of Pollution by Population Types',
         x = 'Pollution Levels',
         y = 'Probability Mass',
         caption = 'Testing Data')
print(lineplot_pmf)
```

Prob Mass Func of Pollution by Population Types



## 1.6 Various Quantiles

Measure quantiles of pollution exposures for different population groups:

1. Consider CDF larger than current quantile of interest.
2. Slice group-specific CDF that is higher and closest to quantile of interest.
3. Merge results for different quantiles together.

```
# Generate pollution quantiles by population groups
df_pop_pollution_distribution <- df_pop_pollution_by_popgrp_cdf %>%
  mutate(pm10_mean = weighted.mean(avgdailypm10, pop_frac)) %>%
  filter(cdf_pop_condi_popgrp_sortpm10 >= 0.10) %>%
  slice(1) %>%
  mutate(pm10_p10 = avgdailypm10) %>%
  select(popgrp, pm10_mean, pm10_p10) %>%
  left_join(df_pop_pollu_gini, by='popgrp') %>%
  left_join(df_pop_pollution_by_popgrp_cdf %>%
    filter(cdf_pop_condi_popgrp_sortpm10 >= 0.20) %>%
    slice(1) %>%
    mutate(pm10_p20 = avgdailypm10) %>%
    select(popgrp, pm10_p20),
    by='popgrp') %>%
  left_join(df_pop_pollution_by_popgrp_cdf %>%
    filter(cdf_pop_condi_popgrp_sortpm10 >= 0.50) %>%
    slice(1) %>%
    mutate(pm10_p50 = avgdailypm10) %>%
    select(popgrp, pm10_p50),
    by='popgrp') %>%
  left_join(df_pop_pollution_by_popgrp_cdf %>%
    filter(cdf_pop_condi_popgrp_sortpm10 >= 0.80) %>%
    slice(1) %>%
    mutate(pm10_p80 = avgdailypm10) %>%
```



```

      select(popgrp, pm10_p80),
      by='popgrp') %>%
left_join(df_pop_pollution_by_popgrp_cdf %>%
      filter(cdf_pop_condi_popgrp_sortpm10 >= 0.90) %>%
      slice(1) %>%
      mutate(pm10_p90 = avgdailypm10) %>%
      select(popgrp, pm10_p90),
      by='popgrp') %>%
select(popgrp, pm10_mean, popgrp_gini, everything())
# display
df_pop_pollution_distribution %>%
  kable(caption = 'PM10 Exposure Distribution by Population Groups') %>%
  kable_styling_fc()

```

PM10 Exposure Distribution by Population Groups

popgrp	pm10_mean	popgrp_gini	pm10_p10	pm10_p20	pm10_p50	pm10_p80	pm10_p90
1	31.07894	0.1206861	24.62676	24.62676	27.64481	30.71275	51.70466
2	39.47897	0.1075096	27.64481	27.64481	31.35114	51.70466	51.70466
3	37.92901	0.1380319	23.56121	30.71275	31.35114	51.70466	54.61304
4	34.86470	0.1543002	19.24456	23.56121	31.35114	51.70466	54.61304
5	32.56731	0.1680265	19.24456	23.56121	30.71275	35.20967	54.61304
6	31.46626	0.1754309	19.24456	23.56121	25.63653	35.20967	54.61304
7	33.50541	0.1453136	19.24456	23.56121	33.98553	45.99021	45.99021