Atkinson Inequality Index and utility Family

Fan Wang

2020-06-12

Contents

1 Atkinson Inequality Index

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 Atkinson Inequality Measures

Atkinson (JET, 1970) studies five standard inequality measures. Atkinson finds that given the same income data across countries, different inequality measure lead to different rankings of which country is more unequal. Atkinson develops an measure of inequality that changes depending on an inequality aversion parameter.

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

$A\left(\left\{Y_i\right\}_{i=1}^N, \lambda\right)$ equals to zero is perfect equality. 1 is Perfect inequality. If $\lambda = 1$, the inequality measure is always equal to 0 because the planner does not care abouot inequality anymore.

1.2 Atkinson Inequality Function

Programming up the equation above, we have:

```
# Formula
ffi_atkinson_ineq <- function(ar_data, fl_rho) {
  ar_data_demean <- ar_data/mean(ar_data)
  it_len <- length(ar_data_demean)
  fl_atkinson <- 1 - sum(ar_data_demean^{fl_rho}*(1/it_len))^(1/fl_rho)
  return(fl_atkinson)
}
```

1.3 Atkinson Inequality Examples

Given a vectr of observables, compute the atkinson inequality measure given different inequality aversion.

Preference vector and data vector:

```
# Preference Vector
ar_rho <- c(1, 1 - (10^(c(seq(-2.2,2.2, length.out=60)))))
ar_rho <- unique(ar_rho)
mt_rho <- matrix(ar_rho, nrow=length(ar_rho), ncol=1)

# Random Data Vector (not equal outcomes)
set.seed(123)
ar_data_rand <- rnorm(15, mean=0,sd=1)
ar_data_rand <- ar_data_rand - min(ar_data_rand) + 1

# Uniform Data Vector (Equal)
ar_data_unif <- rep(1, length(ar_data_rand))

# One Rich (last person has income equal to the sum of all others*100)
ar_data_onerich <- rep(0.1, length(ar_data_rand))
ar_data_onerich[length(ar_data_onerich)] = sum(head(ar_data_onerich,-1))*10
```

Testing Atkinson with different data arrays:

```
# ATK = 0.1180513
ffi_atkinson_ineq(ar_data_rand, -1)
```

```
## [1] 0.1180513
```

```
# ATK = 0
ffi_atkinson_ineq(ar_data_unif, -1)
```

```
## [1] 0
```

```
# ATK = 0.89
ffi_atkinson_ineq(ar_data_onerich, -1)
```

```
## [1] 0.8956933
```

1.3.1 Atkinson Inequality as Inequality Aversion Changes

This is the vector of inequality aversion parameters:

```
ar_rho
```

```
##  [1]    1.00000000    0.99369043    0.99250837    0.99110487    0.98943842    0.98745978    0.985110
##  [9]    0.97900896    0.97507643    0.97040717    0.96486316    0.95828051    0.95046465    0.941184
## [17]    0.91708291    0.90154895    0.88310482    0.86120530    0.83520306    0.80432947    0.767671
## [25]    0.67246762    0.61110666    0.53825013    0.45174443    0.34903248    0.22707813    0.082276
## [33]   -0.29379184   -0.53617495   -0.82396688   -1.16567469   -1.57139912   -2.05313328   -2.625117
## [41]   -4.11063160   -5.06807371   -6.20488608   -7.55467254   -9.15733231  -11.06023949  -13.319643
## [49]  -19.18760255  -22.96961271  -27.46015678  -32.79197376  -39.12267043  -46.63938010  -55.564294
## [57]  -78.74343059  -93.68282046 -111.42100351 -132.48231461 -157.48931925
```

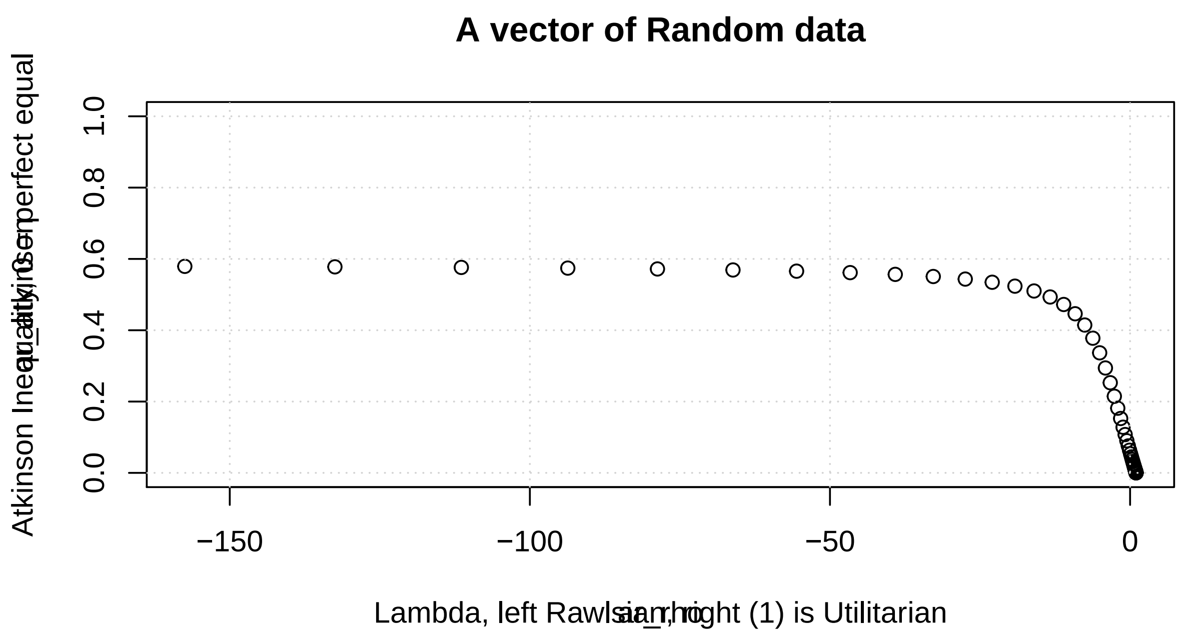
How does Atkinson Inequality measure change with respect to a vector of random data as inequality aversion shifts:

```
par(new=T)
st_x_label <- 'Lambda, left Rawlsian, right (1) is Utilitarian'
```

```
st_y_label <- 'Atkinson Inequality, 0 = perfect equal'
ar_ylim = c(0,1)
ffi_atkinson_ineq(ar_data_rand, -1)
```

```
## [1] 0.1180513
```

```
ar_atkinson <- apply(mt_rho, 1, function(row){ffi_atkinson_ineq(ar_data_rand, row[1])})
plot(ar_rho, ar_atkinson, ylim = ar_ylim)
title(main = 'A vector of Random data', xlab = st_x_label, ylab = st_y_label)
grid()
```

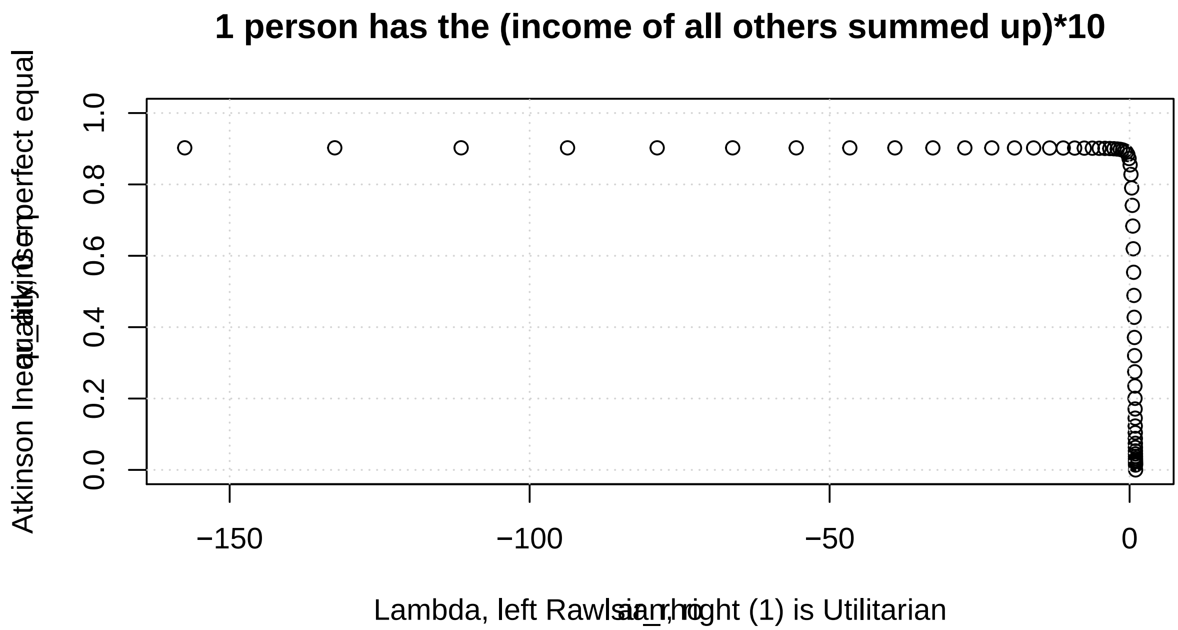


Now with the one person has the wealth of all others in the vector times 10:

```
par(new=T)
ffi_atkinson_ineq(ar_data_onerich, -1)
```

```
## [1] 0.8956933
```

```
ar_atkinson <- apply(mt_rho, 1, function(row){ffi_atkinson_ineq(ar_data_onerich, row[1])})
plot(ar_rho, ar_atkinson, ylim = ar_ylim)
title(main = '1 person has the (income of all others summed up)*10', xlab = st_x_label, ylab = st_y_labe
grid()
```

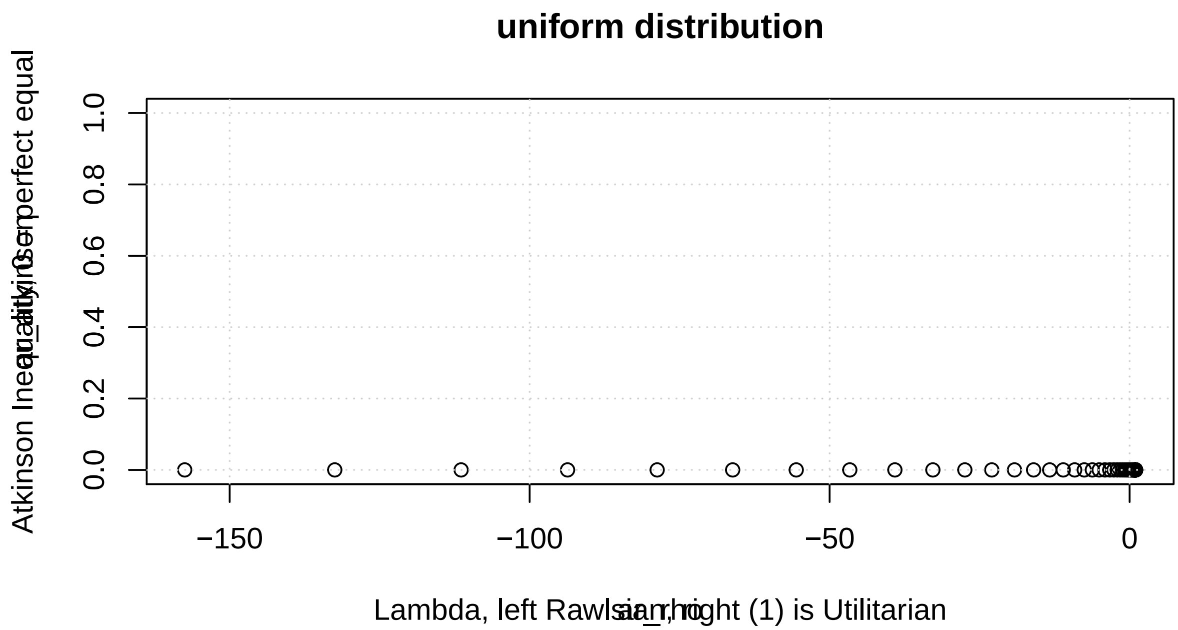


The Uniform Results, since allocations are uniform, zero for all:

```
par(new=T)
ffi_atkinson_ineq(ar_data_unif, -1)
```

```
## [1] 0
```

```
ar_atkinson <- apply(mt_rho, 1, function(row){ffi_atkinson_ineq(ar_data_unif, row[1])})
plot(ar_rho, ar_atkinson, ylim = ar_ylim)
title(main = 'uniform distribution', xlab = st_x_label, ylab = st_y_label)
grid()
```



1.4 Analyzing Equation Mechanics

How does the Aktinson Family utility function work? THe Atkinson Family Utility has the following functional form.

$$V^{\text{social}} = \left(\alpha \cdot A^{\lambda} + \beta \cdot B^{\lambda}\right)^{\frac{1}{\lambda}}$$

Several key issues here:

1. V^{social} is the utility of some social planner
2. A and B are allocations for Alex and Ben.
3. α and β are biases that a social planner has for Alex and Ben: $\alpha + \beta = 1$, $\alpha > 0$, and $\beta > 0$
4. $-\infty < \lambda \leq 1$ is a measure of inequality aversion
   - $\lambda = 1$ is when the planner cares about weighted total allocations (efficient, Utilitarian)
   - $\lambda = -\infty$ is when the planner cares about only the minimum between A and B allocations (equality, Rawlsian)

What if only care about Alex? Clearly, if the planner only cares about Ben, $\beta = 1$, then:

$$V^{\text{social}} = \left(B^{\lambda}\right)^{\frac{1}{\lambda}} = B$$

Clearly, regardless of the value of λ, as B increases V increases. What Happens to V when A or B increases? What is the derivative of V with respect to A or B?

$$\frac{\partial V}{\partial A} = \frac{1}{\lambda}\left(\alpha A^{\lambda} + \beta B^{\lambda}\right)^{\frac{1}{\lambda}-1} \cdot \lambda \alpha A^{\lambda-1}$$

$$\frac{\partial V}{\partial A} = \left(\alpha A^{\lambda} + \beta B^{\lambda}\right)^{\frac{1-\lambda}{\lambda}} \cdot \alpha A^{\lambda-1} > 0$$

Note that $\frac{\partial V}{\partial A} > 0$. When $\lambda < 0$, $Z^{\lambda} > 0$. For example $10^{-2} = \frac{1}{100}$. And For example $0.1^{\frac{3}{-2}} = \frac{1}{0.1^{1.5}}$. Still Positive.

While the overall V increases with increasing A, but if we did not have the outter power term, the situation is different. In particular, when $\lambda < 0$:

$$\text{if } \lambda < 0 \text{ then } \frac{d\left(\alpha A^{\lambda} + \beta B^{\lambda}\right)}{dA} = \alpha \lambda A^{\lambda - 1} < 0$$

Without the outter $\frac{1}{\lambda}$ power, negative λ would lead to decreasing weighted sum. But:

$$\text{if } \lambda < 0 \text{ then } \frac{dG^{\frac{1}{\lambda}}}{dG} = \frac{1}{\lambda} \cdot G^{\frac{1-\lambda}{\lambda}} < 0$$

so when G is increasing and $\lambda < 0$, V would decrease. But when $G(A, B)$ is decreasing, as is the case with increasing A when $\lambda < 0$, V will actually increase. This confirms that $\frac{\partial V}{\partial A} > 0$ for $\lambda < 0$. The result is symmetric for $\lambda > 0$.

1.5 Indifference Curve Graph

Given V^*, we can show the combinations of A and B points that provide the same utility. We want to be able to potentially draw multiple indifference curves at the same time. Note that indifference curves are defined by α, λ only. Each indifference curve is a set of A and B coordinates. So to generate multiple indifference curves means to generate many sets of A, B associated with different planner preferences, and then these could be graphed out.

```
# A as x-axis, need bounds on A
fl_A_min = 0.01
fl_A_max = 3
it_A_grid = 10000

# Define parameters
# ar_lambda <- 1 - (10^(c(seq(-2,2, length.out=3))))
ar_lambda <- c(1, 0.6, 0.06, -6)
ar_beta <- seq(0.25, 0.75, length.out = 3)
ar_beta <- c(0.3, 0.5, 0.7)
ar_v_star <- seq(1, 2, length.out = 1)
tb_pref <- as_tibble(cbind(ar_lambda)) %>%
  expand_grid(ar_beta) %>% expand_grid(ar_v_star) %>%
  rename_all(~c('lambda', 'beta', 'vstar')) %>%
  rowid_to_column(var = "indiff_id")

# Generate indifference points with apply and anonymous function
# tb_pref, whatever is selected from it, must be all numeric
# if there are strings, would cause conversion error.
ls_df_indiff <- apply(tb_pref, 1, function(x){
  indiff_id <- x[1]
  lambda <- x[2]
  beta <- x[3]
  vstar <- x[4]
  ar_fl_A_indiff <- seq(fl_A_min, fl_A_max, length.out=it_A_grid)
  ar_fl_B_indiff <- (((vstar^lambda) -
                        (beta*ar_fl_A_indiff^(lambda)))/(1-beta))^(1/lambda)
  mt_A_B_indiff <- cbind(indiff_id, lambda, beta, vstar,
                         ar_fl_A_indiff, ar_fl_B_indiff)
  colnames(mt_A_B_indiff) <- c('indiff_id', 'lambda', 'beta', 'vstar',
```

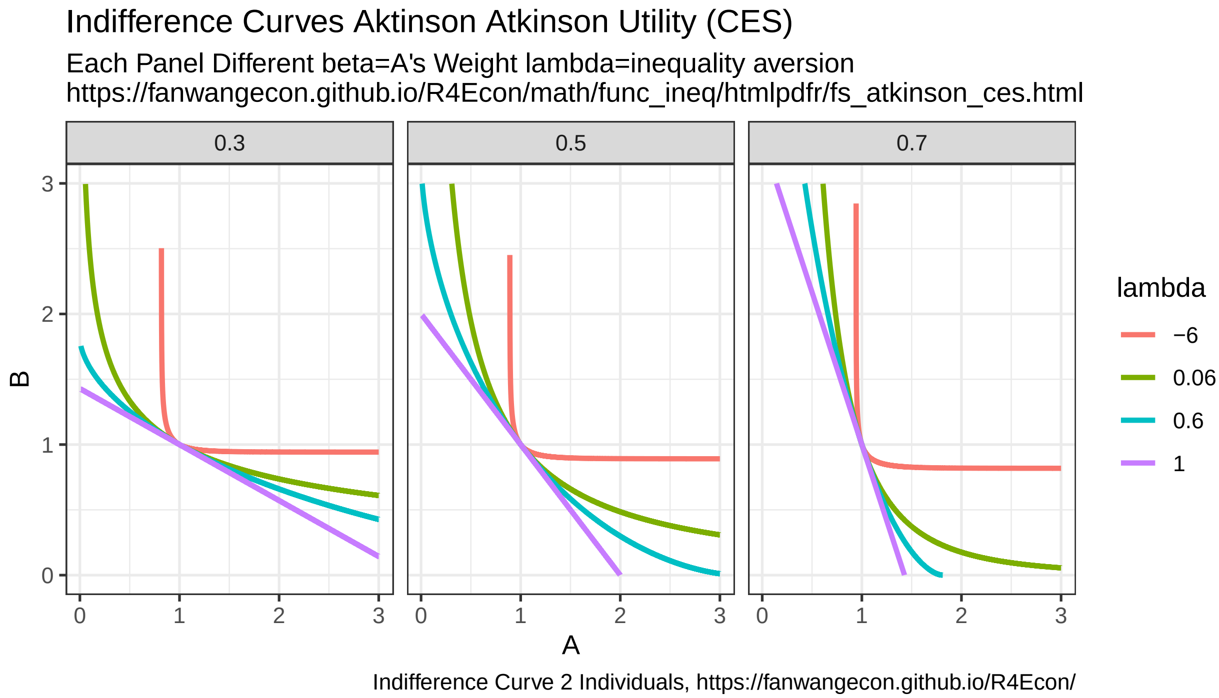
```
                               'indiff_A', 'indiff_B')
  tb_A_B_indiff <- as_tibble(mt_A_B_indiff) %>%
    rowid_to_column(var = "A_grid_id") %>%
    filter(indiff_B >= 0 & indiff_B <= max(ar_fl_A_indiff))
  return(tb_A_B_indiff)
})
df_indiff <- do.call(rbind, ls_df_indiff) %>% drop_na()
```

Note that many more A grid points are needed to fully plot out the leontief line.

```
# Labeling
st_title <- paste0('Indifference Curves Aktinson Atkinson Utility (CES)')
st_subtitle <- paste0('Each Panel Different beta=A\'s Weight lambda=inequality aversion\n',
                      'https://fanwangecon.github.io/',
                      'R4Econ/math/func_ineq/htmlpdfr/fs_atkinson_ces.html')
st_caption <- paste0('Indifference Curve 2 Individuals, ',
                     'https://fanwangecon.github.io/R4Econ/')
st_x_label <- 'A'
st_y_label <- 'B'

# Graphing
plt_indiff <-
  df_indiff %>% mutate(lambda = as_factor(lambda),
                       beta = as_factor(beta),
                       vstar = as_factor(vstar)) %>%
  ggplot(aes(x=indiff_A, y=indiff_B,
             colour=lambda)) +
  facet_wrap( ~ beta) +
  geom_line(size=1) +
  labs(title = st_title, subtitle = st_subtitle,
       x = st_x_label, y = st_y_label, caption = st_caption) +
  theme_bw()

# show
print(plt_indiff)
```


Indifference Curves Aktinson Atkinson Utility (CES)
Each Panel Different beta=A's Weight lambda=inequality aversion
https://fanwangecon.github.io/R4Econ/math/func_ineq/htmlpdfr/fs_atkinson_ces.html
0.3
0.5
0.7
B
A
lambda
−6
0.06
0.6
1
Indifference Curve 2 Individuals, https://fanwangecon.github.io/R4Econ/