Modeling Income Distribution

Let \( \{x_1, \cdots, x_n\} \) denote some sample. Then

\[
\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} \frac{1}{n} x_i
\]

This can be used when we have census data.

```r
load(url("http://freakonometrics.free.fr/income_5.RData"))
income <- sort(income)
plot(1:5, income)
```

It is possible to use survey data. If \( \pi_i \) denote the probability to be drawn, use weights

\[
\omega_i \propto \frac{1}{n\pi_i}
\]
The weighted average is then

\[ \bar{x}_\omega = \sum_{i=1}^{n} \frac{\omega_i}{\omega} x_i \]

where \( \omega = \sum \omega_i \). This is an unbiased estimator of the population mean.

Sometime, data are obtained from stratified samples: before sampling, members of the population are grouped in homogeneous subgroupes (called a strata). Given \( S \) strata, such that the population in strata \( s \) is \( N_s \), then

\[ \bar{x}_S = \sum_{s=1}^{S} \frac{N_s}{N} \bar{x}_s \text{ where } \bar{x}_s = \frac{1}{N_s} \sum_{i \in S_s} x_i \]
Statistical Tools Used to Describe the Distribution

Consider a sample \( \{x_1, \cdots, x_n\} \). Usually, the order is not important. So let us order those values,

\[
\begin{align*}
\mathop{\min}\{x_i\} & \leq x_1:n \leq x_2:n \leq \cdots \leq x_{n-1:n} \leq x_{n:n} \\
\mathop{\max}\{x_i\}
\end{align*}
\]

As usual, assume that \( x_i \)'s were randomly drawn from an (unknown) distribution \( F \).

If \( F \) denotes the cumulative distribution function, \( F(x) = \mathbb{P}(X \leq x) \), one can prove that

\[
F(x_{i:n}) = \mathbb{P}(X \leq x_{i:n}) \sim \frac{i}{n}
\]

The quantile function is defined as the inverse of the cumulative distribution function \( F \),

\[
Q(u) = F^{-1}(u) \text{ or } F(Q(u)) = \mathbb{P}(X \leq Q(u)) = u
\]
Lorenz curve

The empirical version of Lorenz curve is

\[
L = \left\{ \frac{i}{n}, \frac{1}{n\bar{x}} \sum_{j \leq i} x_{j:n} \right\}
\]

```r
1 > plot((0:5)/5, c(0, cumsum(income)/sum(income)))
```
Gini Coefficient

Gini coefficient is defined as the ratio of areas, \( \frac{A}{A + B} \).

It can be defined using order statistics as

\[
G = \frac{2}{n(n-1)\bar{x}} \sum_{i=1}^{n} i \cdot x_{i:n} - \frac{n + 1}{n - 1}
\]

```r
> n <- length(income)
> mu <- mean(income)
> 2*sum((1:n)*sort(income))/ (mu*n*(n-1)) -(n+1)/(n-1)
[1] 0.5800019
```
Distribution Fitting

Assume that we now have more observations,

```r
> load(url("http://freakonometrics.free.fr/income_500.RData"))
```

We can use some histogram to visualize the distribution of the income

```r
> summary(income)

   Min.  1st Qu.   Median      Mean   3rd Qu.  Max.  
 2191.0 23830.0  42750.0  77010.0  87430.0 2003000

> sort(income)[495:500]

   [1] 465354 489734 512231 539103 627292 2003241

> hist(income, breaks=seq(0,2005000,by=5000))
```
Distribution Fitting

Because of the dispersion, look at the histogram of the logarithm of the data

1. \[ \text{hist} \left( \log(\text{income}, 10), \text{breaks} = \text{seq}(3, 6.5, \text{length} = 51) \right) \]
2. \[ \text{boxplot} \left( \text{income}, \text{horizontal} = \text{TRUE}, \log = "x" \right) \]
Distribution Fitting

The cumulative distribution function (on the log of the income)

1. $u \leftarrow \text{sort}(\text{income})$
2. $v \leftarrow (1:500)/500$
3. $\text{plot}(u,v, \text{type}="s", \text{log}="x")$
If we invert that graph, we have the quantile function

```r
> plot(v,u,type="s",col="red",log="y")
```
Distribution Fitting

On that dataset, Lorenz curve is

```r
> plot((0:500)/500, c(0, cumsum(income)/sum(income)))
```
There are two techniques to get the distribution of an estimator $\hat{\theta}$,

- a parametric one, based on some assumptions on the underlying distribution,
- a nonparametric one, based on sampling techniques

If $X_i$’s have a $\mathcal{N}(\mu, \sigma^2)$ distribution, then $X \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right)$

But sometimes, distribution can only be obtained as an approximation, because of asymptotic properties.

From the central limit theorem, $X \to \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right)$ as $n \to \infty$.

In the nonparametric case, the idea is to generate pseudo-samples of size $n$, by resampling from the original distribution.
Bootstrapping

Consider a sample \( x = \{x_1, \cdots, x_n\} \). At step \( b = 1, 2, \cdots, B \), generate a pseudo sample \( x^b \) by sampling (with replacement) within sample \( x \). Then compute any statistic \( \hat{\theta}(x^b) \)

```r
> boot <- function(sample, f, b=500)
+    F <- rep(NA, b)
+    n <- length(sample)
+    for (i in 1:b)
+    { idx <- sample(1:n, size=n, replace=TRUE)
+        F[i] <- f(sample[idx])
+    }
+    return(F)
```
Bootstraping

Let us generate 10,000 bootstrapped sample, and compute Gini index on those

```r
> boot_gini <- boot(income, gini, 1e4)
```

To visualize the distribution of the index

```r
> hist(boot_gini, probability=TRUE)
> u <- seq(.4, .7, length=251)
> v <- dnorm(u, mean(boot_gini), sd(boot_gini))
> lines(u, v, col="red", lty=2)
```
Continuous Versions

The empirical cumulative distribution function

\[ \hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} 1(x_i \leq x) \]

Observe that

\[ \hat{F}_n(x_{j:n}) = \frac{j}{n} \]

If \( F \) is absolutely continuous,

\[ F(x) = \int_{0}^{x} f(t)dt \text{ i.e. } f(x) = \frac{dF(x)}{dx}. \]

Then

\[ \mathbb{P}(x \in [a, b]) = \int_{a}^{b} f(t)dt = F(b) - F(a). \]
Continuous Versions

One can define quantiles as

\[ x = Q(p) = F^{-1}(p) \]

The expected value is

\[
\mu = \int_0^\infty x f(x) \, dx = \int_0^\infty [1 - F(x)] \, dx = \int_0^1 Q(p) \, dp.
\]

We can compute the average standard of living of the group below \( z \). This is equivalent to the expectation of a truncated distribution.

\[
\mu^-_z = \frac{1}{F(z)} \int_0^z x f(x) \, dx = \int_0^\infty \left[ 1 - \frac{F(x)}{F(z)} \right] f(x)
\]
Continuous Versions

Lorenz curve is \( p \mapsto L(p) \) with

\[
L(p) = \frac{1}{\mu} \int_0^{Q(p)} xf(x)dx
\]

Gastwirth (1971) proved that

\[
L(p) = \frac{1}{\mu} \int_0^p Q(u)du = \frac{\int_0^p Q(u)du}{\int_0^1 Q(u)du}
\]

The numerator sums the incomes of the bottom \( p \) proportion of the population. The denominator sums the incomes of all the population.

\( L \) is a \([0, 1] \rightarrow [0, 1]\) function, continuous if \( F \) is continuous. Observe that \( L \) is increasing, since

\[
\frac{dL(p)}{dp} = \frac{Q(p)}{\mu}
\]

Further, \( L \) is convex
The sample case

\[
L \left( \frac{i}{n} \right) = \frac{\sum_{j=1}^{i} x_{j:n}}{\sum_{j=1}^{n} x_{j:n}}
\]

The points \(\{i/n, L(i/n)\}\) are then linearly interpolated to complete the corresponding Lorenz curve.

The continuous distribution case

\[
L(p) = \frac{\int_{0}^{F^{-1}(p)} ydF(y)}{\int_{0}^{\infty} ydF(y)} = \frac{1}{E(X)} \int_{0}^{p} F^{-1}(u)du
\]

with \(p \in (0, 1)\).

Let \(L\) be a continuous function on \([0, 1]\), then \(L\) is a Lorenz curve if and only if

\[
L(0) = 0, \ L(1) = 1, \ L'(0^+) \geq 0 \text{ and } L''(p) \geq 0 \text{ on } [0, 1].
\]
From Lorenz to Bonferroni

The Bonferroni curve is

\[ B(p) = \frac{L(p)}{p} \]

and the Bonferroni index is

\[ BI = 1 - \int_0^1 B(p) dp. \]

Define

\[ P_i = \frac{i}{n} \text{ and } Q_i = \frac{1}{n\bar{x}} \sum_{j=1}^{i} x_j \]

then

\[ B = \frac{1}{n - 1} \sum_{i=1}^{n-1} \left( \frac{P_i - Q_i}{P_i} \right) \]
Gini and Pietra indices

The Gini index is defined as twice the area between the egalitarian line and the Lorenz curve

\[ G = 2 \int_0^1 [p - L(p)] dp = 1 - 2 \int_0^1 L(p) dp \]

which can also be written

\[ 1 - \frac{1}{\mathbb{E}(X)} \int_0^\infty [1 - F(x)]^2 dx \]

Pietra index is defined as the maximal vertical deviation between the Lorenz curve and the egalitarian line

\[ P = \max_{p \in (0,1)} \{ p - L(p) \} = \frac{\mathbb{E}(|X - \mathbb{E}(X)|)}{2\mathbb{E}(X)} \]

if \( F \) is strictly increasing (the maximum is reached in \( p^* = F(\mathbb{E}(X)) \))
Examples

E.g. consider the uniform distribution

\[ F(x) = \min\{1, \frac{x-a}{b-a} 1(x \geq a)\} \]

Then

\[ L(p) = \frac{2ap + (b-a)^2p^2}{a+b} \]

and Gini index is

\[ G = \frac{b-a}{3(a+b)} \]

E.g. consider a Pareto distribution,

\[ F(x) = 1 - \left(\frac{x_0}{x}\right)^\alpha, \ x \geq x_0, \]

with shape parameter \( \alpha > 0 \). Then

\[ F^{-1}(u) = \frac{x_0}{(1-u)^{\frac{1}{\alpha}}} \]
and
\[ L(p) = 1 - [1 - p]^{1 - \frac{1}{\alpha}} \quad p \in (0, 1). \]

and Gini index is
\[ G = \frac{1}{2\alpha - 1} \]

while Pietra index is, if \( \alpha > 1 \)
\[ P = \frac{(\alpha - 1)^{\alpha - 1}}{\alpha^\alpha} \]

E.g. consider the lognormal distribution,
\[ F(x) = \Phi \left( \frac{\log x - \mu}{\sigma} \right) \]
then
\[ L(p) = \Phi(\Phi^{-1}(p) - \sigma) \quad p \in (0, 1). \]
and Gini index is
\[ G = 2\Phi \left( \frac{\sigma}{\sqrt{2}} \right) - 1 \]
Fitting a Distribution

The standard technique is based on maximum likelihood estimation, provided by

```r
> library(MASS)
> fitdistr(income, "lognormal")

meanlog   sdlog
10.72264538 1.01091329
( 0.04520942) ( 0.03196789)
```

For other distribution (such as the Gamma distribution), we might have to rescale

```r
> (fit_g <- fitdistr(income/1e2, "gamma"))

shape   rate
1.0812757769 0.0014040438
(0.0473722529) (0.0000544185)

> (fit_ln <- fitdistr(income/1e2, "lognormal"))

meanlog   sdlog
6.11747519 1.01091329
(0.04520942) (0.03196789)
```
Fitting a Distribution

We can compare the densities

```r
> u = seq(0, 2e5, length = 251)
> hist(income, breaks = seq(0, 2005000, by = 5000),
  col = rgb(0, 0, 1, .5), border = "white", xlim = c(0, 2e5), probability = TRUE)
> v_g <- dgamma(u/1e2, fit_g$estimate[1], fit_g$estimate[2]) / 1e2
> v_ln <- dlnorm(u/1e2, fit_ln$estimate[1], fit_ln$estimate[2]) / 1e2
> lines(u, v_g, col = "red", lty = 2)
> lines(u, v_ln, col = rgb(1, 0, 0, .4))
```
Fitting a Distribution

or the cumulative distributions

```r
x <- sort(income)
y <- (1:500)/500
plot(x, y, type="s", col="black")
v_g <- pgamma(u/1e2, fit_g$estimate[1], fit_g$estimate[2])
v_ln <- plnorm(u/1e2, fit_ln$estimate[1], fit_ln$estimate[2])
lines(u, v_g, col = "red", lty = 2)
lines(u, v_ln, col = rgb(1, 0, 0, .4))
```

One might consider the parametric version of Lorenz curve, to confirm the goodness of fit, e.g. a lognormal distribution with $\sigma = 1$ since

```r
> fitdistr(income, "lognormal")

<table>
<thead>
<tr>
<th>meanlog</th>
<th>sdlog</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.72264538</td>
<td>1.01091329</td>
</tr>
</tbody>
</table>
```
Fitting a Distribution

We can use functions of R

```r
library(ineq)
Lc.sim <- Lc(income)
plot(0:1,0:1,xlab="p",ylab="L(p)",col="white")
polygon(c(0,1,1,0),c(0,0,1,0),col=rgb(0,0,1,.1),border=NA)
polygon(Lc.sim$p,Lc.sim$L,col=rgb(0,0,1,.3),border=NA)
lines(Lc.sim)
segments(0,0,1,1)
lines(Lc.lognorm, parameter=1,lty=2)
```
Standard Parametric Distribution

For those distributions, we mention the R names in the `gamlss` package. Inference can be done using

```r
fit <- gamlss(y~1, family=LNO)
```

- **log normal**

\[
f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}, \ x \geq 0
\]

with mean \( e^{\mu + \sigma^2/2} \), median \( e^{\mu} \), and variance \( (e^{\sigma^2} - 1)e^{2\mu + \sigma^2} \)

```r
LNO(mu.link = "identity", sigma.link = "log")
dLNO(x, mu = 1, sigma = 0.1, nu = 0, log = FALSE)
```

- **gamma**
\[ f(x) = \frac{x^{1/\sigma^2-1} \exp\left[-x/(\sigma^2 \mu)\right]}{(\sigma^2 \mu)^{1/\sigma^2} \Gamma(1/\sigma^2)}, \quad x \geq 0 \]

with mean \( \mu \) and variance \( \sigma^2 \)

\begin{verbatim}
1 GA(mu.link = "log", sigma.link ="log")
2 dGA(x, mu = 1, sigma = 1, log = FALSE)
\end{verbatim}

- Pareto

\[ f(x) = \frac{\alpha x_m x^\alpha}{x^{\alpha+1}} \quad \text{for} \quad x \geq x_m \]

with cumulated distribution

\[ F(x) = 1 - \left(\frac{x_m}{x}\right)^\alpha \quad \text{for} \quad x \geq x_m \]

with mean \( \frac{\alpha x_m}{(\alpha - 1)} \) if \( \alpha > 1 \), and variance \( \frac{x_m^2 \alpha}{(\alpha - 1)^2(\alpha - 2)} \) if \( \alpha > 2 \).
1 \texttt{PARETO2(mu.\ link = "log", sigma.\ link = "log")}
2 \texttt{dPARETO2(x, mu = 1, sigma = 0.5, log = FALSE)}
Larger Families

- **GB1** - generalized Beta type 1

\[
f(x) = \frac{|a|x^{ap-1}(1 - (x/b)^a)^{q-1}}{b^{ap}B(p,q)}, \quad 0 < x^a < b^a
\]

where \( b, p, \) and \( q \) are positive

```r
1  GB1(mu.link = "logit", sigma.link = "logit", nu.link = "log",tau.link = "log")
2  dGB1(x, mu = 0.5, sigma = 0.4, nu = 1, tau = 1, log = FALSE)
```

The GB1 family includes the generalized gamma(GG), and Pareto as special cases.

- **GB2** - generalized Beta type 2
\[ f(x) = \frac{|a|x^{ap-1}}{b^{ap}B(p, q)(1 + (x/b)^a)^{p+q}} \]

The GB2 nests common distributions such as the generalized gamma (GG), Burr, lognormal, Weibull, Gamma, Rayleigh, Chi-square, Exponential, and the log-logistic.

- Generalized Gamma

\[ f(x) = \frac{(p/a^d)x^{d-1}e^{-(x/a)^p}}{\Gamma(d/p)}, \]
Dealing with Binned Data

```r
> load(url("http://freakonometrics.free.fr/income_binned.RData"))
> head(income_binned)

low  high  number mean   std_err
 1   0    4999   95  3606    964
 2  5000   9999  267  7686   1439
 3 10000  14999  373 12505   1471
 4 15000  19999  350 17408   1368
 5 20000  24999  329 22558   1428
 6 25000  29999  337 27584   1520

> tail(income_binned)

low  high  number mean   std_err
 46 225000 229999   10 228374   1197
 47 230000 234999   13 232920   1370
 48 235000 239999   11 236341   1157
 49 240000 244999   14 242359   1474
 50 245000 249999   11 247782   1487
 51 250000 Inf     228 395459  189032
```
Dealing with Binned Data

There is a dedicated package to work with such datasets,

```r
> library(binequality)
```

To fit a parametric distribution, e.g. a log-normal distribution, use functions of `R`

```r
> n <- nrow(income_binned)
> fit_LN <- fitFunc(ID=rep("Fake Data",n), hb=income_binned[, "number"],
                   bin_min=income_binned[, "low"], bin_max=income_binned[, "high"],
                   obs_mean=income_binned[, "mean"], ID_name="Country",
                   distribution=LNO, distName="LNO")
```

Time difference of 0.09900618 secs

for LNO fit across 1 distributions
Dealing with Binned Data

To visualize the cumulated distribution function, use

```r
> N <- income_binned$number
> y1 <- cumsum(N)/sum(N)
> u <- seq(min(income_binned$low), max(income_binned$low), length=101)
> v <- plnorm(u, fit_LN$parameters[1], fit_LN$parameters[2])
> plot(u, v, col="blue", type="l", lwd=2, xlab="Income", ylab="Cumulative Probability")
> for (i in 1:(n-1)) rect(income_binned$low[i], 0, income_binned$high[i], y1[i], col=rgb(1,0,0,.2))
> for (i in 1:(n-1)) rect(income_binned$low[i], y1[i], income_binned$high[i], c(0, y1)[i], col=rgb(1,0,0,.4))
```
Dealing with Binned Data

and to visualize the cumulated distribution function, use

```r
> N = income_binned$number
> y2 = N / sum(N) / diff(income_binned$low)
> u = seq(min(income_binned$low), max(income_binned$low), length = 101)
> v = dlnorm(u, fit_LN$parameters[1], fit_LN$parameters[2])
> plot(u, v, col = "blue", type = "l", lwd = 2, xlab = "Income", ylab = "Density")
> for (i in 1:(n-1)) rect(income_binned$low[i], 0, income_binned$high[i], y2[i], col = rgb(1, 0, 0, .2), border = "white")
```
Dealing with Binned Data

But it is also possible to estimate all GB-distributions at once,

```r
> fits = run_GB_family (ID = rep("Fake Data", n), hb = income_binned[, "number"],
                     bin_min = income_binned[, "low"], bin_max = income_binned[, "high"], obs_mean = income_binned[, "mean"],
                     + ID_name = "Country")
```

Time difference of 0.03800201 secs for GB2 fit across 1 distributions

Time difference of 0.3090181 secs for GG fit across 1 distributions

Time difference of 0.864049 secs for BETA2 fit across 1 distributions

...
for LOGLOG fit across 1 distributions

Time difference of 1.865106 secs
for PARETO2 fit across 1 distributions

```r
fits$fit.filter[, c("gini", "aic", "bic")]
gini   aic   bic
   1   NA   NA   NA
   2  5.054377 34344.87 34364.43
   3  5.110104 34352.93 34372.48
   4   NA  53638.39  53657.94
   5  4.892090 34845.87  34865.43
   6  5.087506 34343.08  34356.11
   7  4.702194 34819.55  34832.59
   8  4.557867 34766.38  34779.41
   9   NA  58259.42  58272.45
  10  5.244332 34805.70  34818.73
```

```r
> fits$best_model$aic
```
<table>
<thead>
<tr>
<th>Country</th>
<th>obsMean</th>
<th>distribution</th>
<th>estMean</th>
<th>var</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fake Data</td>
<td>NA</td>
<td>LNO</td>
<td>72328.86</td>
<td>6969188937</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>cv</th>
<th>cv_sqr</th>
<th>gini</th>
<th>theil</th>
<th>MLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.154196</td>
<td>1.332168</td>
<td>5.087506</td>
<td>0.4638252</td>
<td>0.4851275</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>aic</th>
<th>bic</th>
<th>didConverge</th>
<th>logLikelihood</th>
<th>nparams</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34343.08</td>
<td>34356.11</td>
<td>TRUE</td>
<td>−17169.54</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>median</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>44400.23</td>
<td>83481.67</td>
</tr>
</tbody>
</table>

That was easy, those were simulated data...
Dealing with Binned Data

Consider now some real data,

```r
> data = read.table("http://freakonometrics.free.fr/us_income.txt", sep=" ", header=TRUE)
> head(data)

  low high number_1000s  mean  std_err
  1   0  4999          4245   1249    50
  2  5000  9999          5128   7923    30
  3 10000 14999          7149  12389    28
  4 15000 19999          7370  17278    26

> tail(data)

  low high number_1000s  mean  std_err
  39 190000 194999        361  192031    115
  40 195000 199999        291  197120    135
  41 200000 249999        2160 219379    437
  42 250000 999999         2498 398233    6519
```
Dealing with Binned Data

To fit a parametric distribution, e.g. a log-normal distribution, use

```r
> n <- nrow(data)
> fit_LN <- fitFunc(ID=rep("US", n), hb=data[, "number_1000s"], bin_min=data[, "low"], bin_max=data[, "high"], obs_mean=data[, "mean"], ID_name="Country", distribution=LNO, distName="LNO")
```

Time difference of 0.1040058 secs

for LNO fit across 1 distributions
Dealing with Binned Data

To visualize the cumulated distribution function, use

```r
> N <- income_binned$number
> y1 <- cumsum(N)/sum(N)
> u <- seq(min(income_binned$low), max(income_binned$low), length=101)
> v <- plnorm(u, fit_LN$parameters[1], fit_LN$parameters[2])
> plot(u, v, col="blue", type="l", lwd=2, xlab="Income", ylab="Cumulative Probability")
> for (i in 1:(n-1)) rect(income_binned$low[i], 0, income_binned$high[i], y1[i], col=rgb(1,0,0,.2))
> for (i in 1:(n-1)) rect(income_binned$low[i], y1[i], income_binned$high[i], c(0,y1)[i], col=rgb(1,0,0,.4))
```
Dealing with Binned Data

and to visualize the cumulated distribution function, use

```r
> N = income_binned$number
> y2 = N / sum(N) / diff(income_binned$low)
> u = seq(min(income_binned$low), max(income_binned$low), length = 101)
> v = dlnorm(u, fit_LN$parameters[1], fit_LN$parameters[2])
> plot(u, v, col = "blue", type = "l", lwd = 2, xlab = "Income", ylab = "Density")
> for (i in 1:(n - 1)) rect(income_binned$low[i], 0, income_binned$high[i], y2[i], col = rgb(1, 0, 0, .2), border = "white")
```
Dealing with Binned Data

And the winner is....

```r
> fits$fit.filter[, c("gini", "aic", "bic")]

<table>
<thead>
<tr>
<th></th>
<th>gini</th>
<th>aic</th>
<th>bic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.413411</td>
<td>825368.7</td>
<td>825407.4</td>
</tr>
<tr>
<td>2</td>
<td>4.395078</td>
<td>825598.8</td>
<td>825627.9</td>
</tr>
<tr>
<td>3</td>
<td>4.455112</td>
<td>825502.4</td>
<td>825531.5</td>
</tr>
<tr>
<td>4</td>
<td>4.480844</td>
<td>825881.5</td>
<td>825910.6</td>
</tr>
<tr>
<td>5</td>
<td>4.413282</td>
<td>825315.3</td>
<td>825344.4</td>
</tr>
<tr>
<td>6</td>
<td>4.922123</td>
<td>832408.2</td>
<td>832427.6</td>
</tr>
<tr>
<td>7</td>
<td>4.341085</td>
<td>827065.2</td>
<td>827084.6</td>
</tr>
<tr>
<td>8</td>
<td>4.318694</td>
<td>826112.9</td>
<td>826132.2</td>
</tr>
<tr>
<td>9</td>
<td>NA</td>
<td>831054.2</td>
<td>831073.6</td>
</tr>
<tr>
<td>10</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

> fits$best_model$aic

<table>
<thead>
<tr>
<th>Country</th>
<th>obsMean</th>
<th>distribution</th>
<th>estMean</th>
<th>var</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>NA</td>
<td>GG</td>
<td>65147.54</td>
<td>3152161910</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>cv</th>
<th>cv_sqr</th>
<th>gini</th>
<th>theil</th>
<th>MLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8617995</td>
<td>0.7426984</td>
<td>4.395078</td>
<td>0.3251443</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>aic</th>
<th>bic</th>
<th>didConverge</th>
<th>logLikelihood</th>
<th>nparams</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>825598.8</td>
<td>825627.9</td>
<td>TRUE</td>
<td>−412796.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>median</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>48953.6</td>
</tr>
</tbody>
</table>