Technical Report: Visualizing Global Income Distributions

Andrew Whitby, Tariq Khokhar, Espen Beer-Prydz, Umar Serajuddin

July 19, 2017

Abstract

The distribution of wealth and income is an issue of particular recent public, as well as long-term academic, interest. While much discussion of distribution is anecdotal, objective data on income distribution is available from sources such as censuses, income surveys and taxation records.

The World Bank hosts a substantial public database of income distribution information in the form of PovCalNet. PovCalNet is primarily designed as an online interface to replicate the Bank's calculations on the incidence of extreme poverty globally. As a by-product, it provides information on the income (or consumption) distributions of many of the world's economies at multiple points in time. However, in general this information is not easily accessible—a point of criticism from others in the development community.

The objective of this project was to make access to this data easier, exposing the distributional component of PovCalNet source data, and allowing visual manipulation of income distributions. This report details the various stages of data extraction, modelling and visualization that were undertaken to produce the prototype web app, which is available at:

http://econandrew.github.io/visual-income-distributions/

The project is very much still a work in progress, and so this report is mostly intended to capture issues, trade-offs and initial design decisions, rather than a set of best practices - though in time we hope it may evolve to do the latter.

Contents

Chapter 1

Introduction

Note on this report

This report is written as a series of Jupyter Notebooks, a form of document that allows execution of scientific and technical Python code. All the code samples are executable, so others can explore and adapt this work online.

In print, the code samples may be hidden. To view them, see the online version of these Notebooks at

http://etc.etc

The distribution of wealth and income is an issue of particular recent public, as well as long-term academic, interest. While much discussion of distribution is anecdotal, objective data on income distribution is available from sources such as censuses, income surveys and taxation records. (Objective and comprehensive information of the distribution of wealth is generally more difficult to obtain, and is not the focus of this project.)

The World Bank hosts a substantial public database of income distribution information in the form of PovCalNet. PovCalNet is primarily designed as an online interface to replicate the Bank's calculations on the incidence of extreme poverty globally. As a by-product, it provides information on the income (or consumption) distributions of many of the world's economies at multiple points in time. However, in general this information is not easily accessible---a point of criticism from others in the development community.

The objective of this project was to make access to this data easier, exposing the distributional component of PovCalNet source data, and allowing visual manipulation of income distributions. Figure 1.1 shows a screenshot of the prototype app in use.

1.1 Previous efforts to visualise global income distributions

There is a long academic and policy literature involving the construction and visualisation of, and generation of statistics from, global income distributions. For example, ...

Few efforts have targeted a lay audience. Notable amongst these is the the "mountain chart" app from Gapminder, which allows the visualisation of a stacked income density for countries on a log income axis. The app uses a two-parameter lognormal approximation, so interesting variation in the shapes of different distributions is lost (beyond scale and spread). This choice does, however, enable Gapminder to provide estimations of income distributions back to 1800, using the method of Zandel *et al*, requiring only estimates of GDP per capita, Gini index and population for each country-year.

The "Our World in Data" provides a similar, though less detailed, view of the same data as part of a broader discussion of global inequality.

Recently, during the course of this project, the World Wealth & Income Database (formerly World Top Incomes Database) expanded the range of visualisations available on their website, focusing particularly on the evolution of inequality statistics.
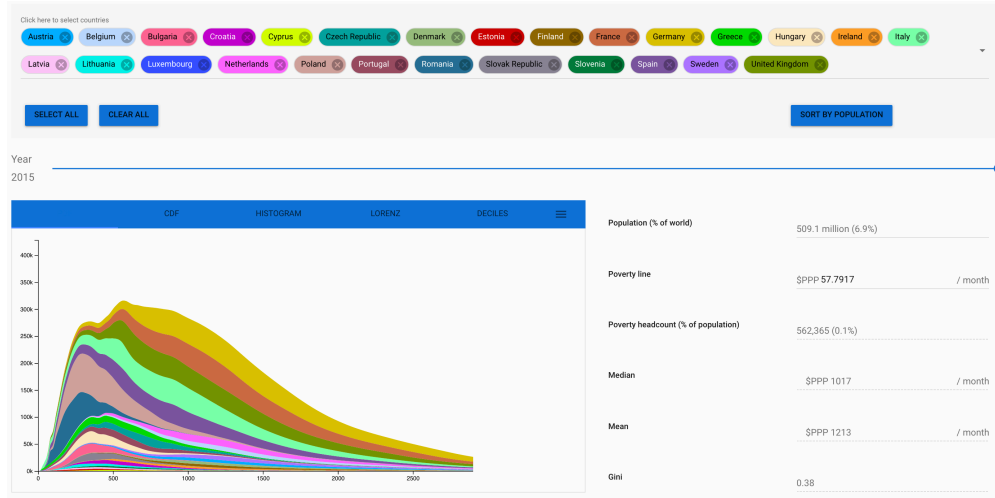
Figure 1.1: *Screenshot of the proof-of-concept web app, showing the aggregate distribution of the EU28 countries in 2015 (excluding Malta which is not in the PovCalNet collection)*
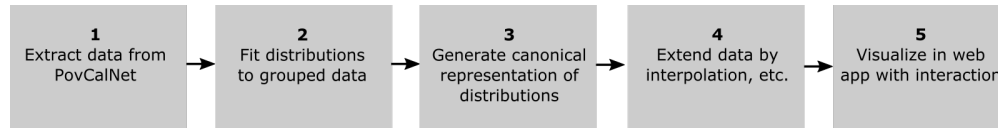

Figure 1.2: *Data and modelling pipeline of the project*

1.2   Objectives of this project

Compared with previous work described above, we set out to:

1. **Maintain a degree of comparability with PovCalNet and the World Bank's global poverty estimates.** This implied that he distributions should remain close to the survey data itself. So, rather than collapsing each distribution to two statistics (mean and Gini index), we desired to maintain a much greater degree of detail of the original distribution as used within PovCalNet, suggesting non-parametric methods of representing the distributions. Moreover methods of interpolation, extrapolation and aggregation should be consistent with those of PovCalNet if possible.

2. **Test a variety of different distribution visualisations.** Probability densities, while quite standard in the academic literature, may not be the most effective way to communicate distribution to non-experts. National statistical organizations, for example, often show histograms instead. Decile shares or averages - or even simple quantile ratios may be more immediately meaningful to users. This is an empirical question, but to examine it we need to capacity to visualise distributions in more than one way. Therefore distributions must be represented internally in such a way as to make this possible.

1.3   Structure of this report

This report is broke into five main chapters (with chapter 2 split into two parts), reflecting the major stages in the data and modelling pipeline from PovCalNet to our online visualization (Figure 1.2).

Chapter 2

Extracting data from PovCalNet

PovCalNet is primarily designed for generating poverty and distributional statistics for custom groupings of countries and years. It does not, at present, provide an easy way to extract machine-readable data on the underlying distributions. In order to make this possible, we constructed a script to do this.
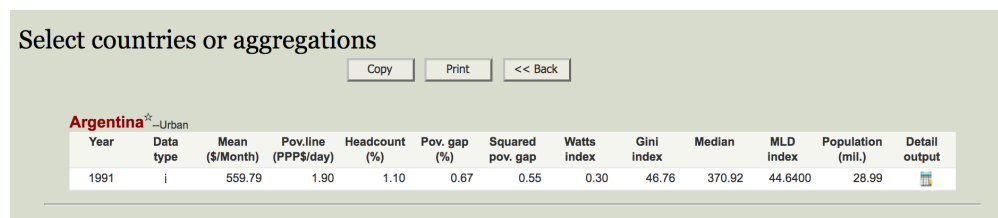
2.1 PovCalNet raw data

PovCalNet contains data on income distributions for TODO country-year or country-part-year combinations. Underlying each of these is distribution data in one of two formats:

- **Unit record**. A vector of raw income (consumption) observations from an actual survey. This is the most common case, particularly for recent years in which survey availability has expanded. Typically this will have a sample size of several thousand, but ranges from under 1000 to over 100,000. For these datasets, statistics like headcount poverty and Gini index are calculated directly from the observations.
- **Grouped**. A list of points on a Lorenz curve, the exact number depending on the detail at which grouped data were available. This format is most common for older data (especially pre-1990), where original survey observations are not available. It is also used for certain countries (e.g. China) where only grouped data are made available to the PovCalNet team. For these datasets, two parametric Lorenz curve models are fitted to the points, and statistics are calculated from the better-fitting distribution.

2.2 The detailed PovCalNet output

PovCalNet is designed to provide formatted output on a particular country-year query, as in Figure 2.1.

However via the "Detail output" link at the right of the table, a far more detailed output can be obtained. For one of the shorter "grouped" format surveys, the first few lines look like this, with data following:

Select countries or aggregations

| Copy | Print | << Back |

Argentina☆–Urban

Year	Data type	Mean ($/Month)	Pov.line (PPP$/day)	Headcount (%)	Pov. gap (%)	Squared pov. gap	Watts index	Gini index	Median	MLD index	Population (mil.)	Detail output
1991	i	559.79	1.90	1.10	0.67	0.55	0.30	46.76	370.92	44.6400	28.99	🏛

Figure 2.1: PovCalNet output for Argentina 1991

```
********************************************************************************
**                          Basic Information                                **
********************************************************************************

---------------- Dataset Information ----------------
                        Country: Argentina
                   Country dode: ARG
                      Data Year: 1991
                       Coverage: Urban
            Welfare measurement: Income
                    Data format: Unit record
                    Data source: ARG_U1991Y
                 Data time span: UnDefined
-----------------------------------------------------
```

Crucially this detailed output contains

- a specfication of the Lorenz curve: either the original points themselves (for grouped data cases) or a gridded sample of 100 points (for unit record cases).
- the survey mean (in local currency and PPP$)

This is enough information to reconstruct the survey data distribution with useably high accuracy.

However, while this detailed output contains much useful information, it is not in a useful format to be ingested automatically by another process.

2.3   Detailed PovCalNet output: as JSON

We use a scraping (text parsing) technique to read the individual data items output in the detailed output and re-output them as JSON (JavaScript Object Notation), as flexible text-based format which can be read by a variety of tools and programming libraries.

The first few lines of the output for the above example looks different, with more of the structure made explicit, but the same values can be seen:

```
{
    "dataset": {
        "source": "ARG_U1991Y",
        "timespan": "UnDefined",
        "coverage": "Urban",
        "year": 1991,
        "iso3c": "ARG",
        "format": "Unit record",
        "country": "Argentina",
        "measure": "Income"
    },
```

This forms the input to the modelling stage of the project.

Chapter 3

Working with grouped income data

Grouped data are unavoidable if one wishes to study global income distributions. We face two sets of cases: in some cases, unit record data are unavailable even to the PovCalNet team (e.g. China); in (many) others, the PovCalNet team has unit record data but cannot share it widely - hence it cannot be the basis for an open visualisation tool. Moreover, PovCalNet automatically outputs grouped data (in Lorenz curve form) regardless of the original input data format (grouped or unit record). For this reason, we take grouped data as the lowest common denominator for distributional data.

Grouped data can take many forms, many of which are equivalent. The basic idea is that all members of the population or sample are listed in ascending order by income. Then, each record represents a contiguous range of these individuals (usually specified as a quantile range) along with a measure of the group's combined income (usually as the sum of their incomes, or their average income, or the sum of their incomes as a share of the population/sample total income). In the usual case, where these groups partition the entire sample/population, these measures may be cumulated.

Additionally, group thresholds may be reported, that is, the minimum and maximum income in each group.

Finally, statistics may be reported for the entire sample/population: the mean, median, minimum, maximum, sample/population size.

The example below generates a population of 150 lognormally distributed incomes, and calculates grouped income along with all of these measures. You will see that the Lorenz cuve ($L(p)$) arises naturally out of the cumulative group statistics.

```
Out[2]:    N  cum_N  Y_min  Y_max  Y_mean  cum_Y_mean  Y_sum  cum_Y_sum  $p$   Y_p  \
        0  15     15   26.0   46.0    37.0        37.0  549.0      549.0  0.1  0.03
        1  60     75   46.0   95.0    69.0        63.0  4146.0     4695.0  0.5  0.25
        2  30    105   95.0  122.0   107.0        75.0  3212.0     7907.0  0.7  0.20
        3  30    135  125.0  185.0   148.0        91.0  4445.0    12352.0  0.9  0.27
        4  15    150  187.0  630.0   266.0       109.0  3989.0    16341.0  1.0  0.24

           cum_Y_p ($L$)
        0           0.03
        1           0.29
        2           0.48
        3           0.76
        4           1.00
```

In most cases not all these measures are available - most can be deduced from a minimal set by simple transformations. Note, however, that this is not the case for group thresholds, which cannot be deduced from other information. Certain fitting methods depend on these thresholds and so will useable when this information is not available. PovCalNet does not currently output group thresholds.
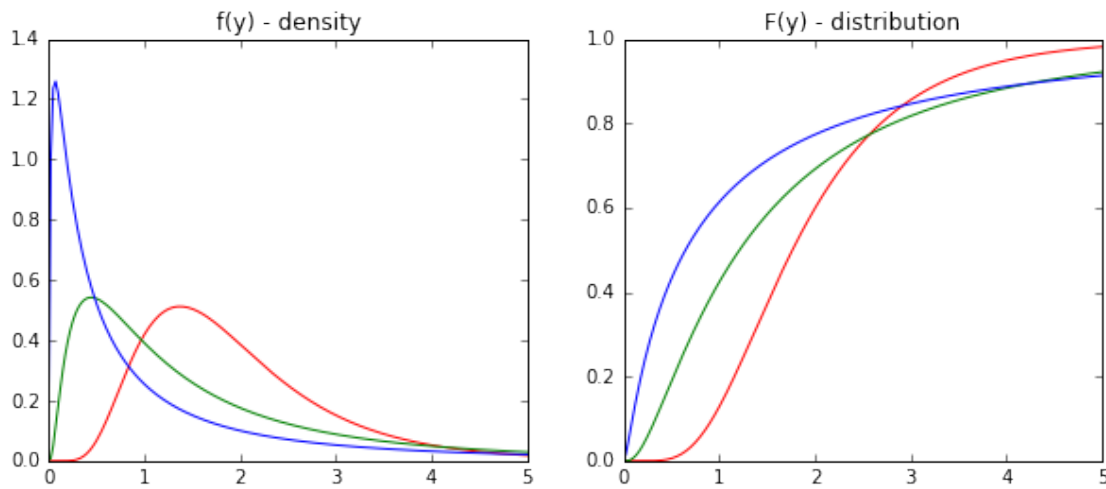
3.1 Lorenz curves

3.1.1 Distributions and densities

Define the cumulative distribution function of income as $F(y)$ so that the proportion of the population having income less than or equal to x is $F(x)$. Then the probability density function $f(y) = d/dy \, F(y)$.

We can take, as an example, the lognormal distribution, which is commonly used to model income distributions. A lognormal random variable is one whose natural logarithm has a normal distribution, hence its pdf and cdf are

$$f(y) = \frac{1}{\sqrt{2\sigma^2}} \exp\left[\frac{(\ln y - \mu)^2}{2\sigma^2}\right] \qquad F(y) = \frac{1}{\sqrt{2\sigma^2}} \int_{-\infty}^{y} \exp\left[\frac{(\ln t - \mu)^2}{2\sigma^2}\right] dt$$

Here's what these two functions look like, with a variety of values of σ and μ chosen so that the mean of the distribution is the same in all cases (2.0):

3.1.2 Constructing Lorenz curves (theory)

Observe that either the density or the (cumulative) distribution function completely describes a probability distribution. A Lorenz curve is another way to describe a probability distribution, up to a scale factor. For each p, $L(p)$ is the fraction of total income earned by the poorest p fraction of the population. More formally

$$L(p) = \frac{\int_0^p Q(z)\,dz}{\int_0^1 Q(z)\,dz} = \frac{\int_0^{Q(p)} y f(y)\,dy}{\mathsf{E}(y)}$$
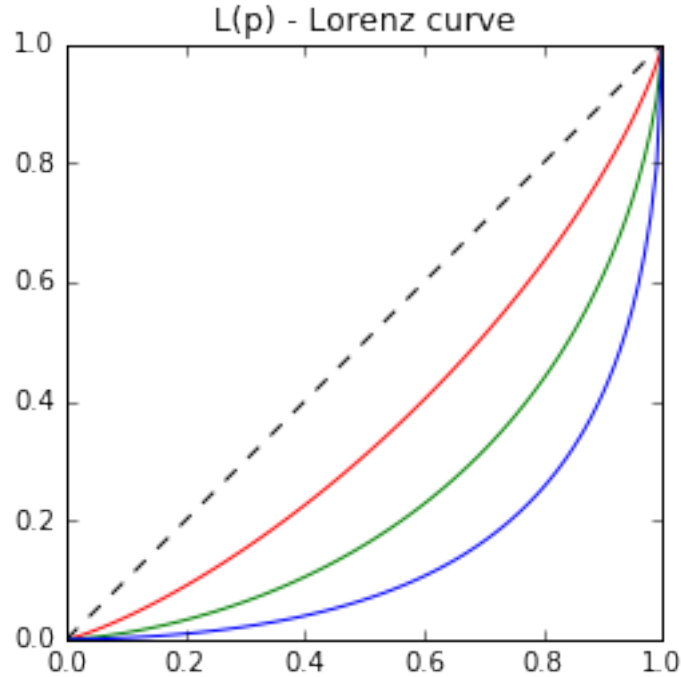
where we define the (generalized) inverse c.d.f., the quantile function,

$$Q(z) = F^{-1}(z) = \inf\{x : F(x) \geq z\}$$

,

and note that $\int_0^p Q(z)\,dz = \int_0^{Q(p)} y f(y)\,dy$. This is the partial mean up to $Q(p)$, which equals the mean income, $\mathsf{E}(y)$, when $p = 1$.

Since $L(p)$ is a constant times the integral of an increasing function $Q(\cdot)$, it is increasing and convex.

The Lorenz curves for the lognormal distributions shown above look like this:

You can observe a few things from this chart. The dashed 45 degree line represents perfect equality: the first 10% of the population earns 10% of the income, etc. The further a Lorenz curve is from that line, the higher the inequality (by this measure).

The blue line, which is for the distribution with the highest variance (σ^2) parameter, also has the highest inequality. And all the Lorenz curves are symmetric about the anti-diagonal (not plotted). These are properties of the lognormal distribution.

3.1.3 Deriving the c.d.f from the Lorenz curve

It follows that given $L(p)$ we can derive the quantile function (inverse c.d.f) as

$$Q(p) = \mathsf{E}(y) \cdot L'(p)$$

where $L'(p) = \frac{d}{dp}L(p)$.

Since the support of a distribution is given by $[Q(0), Q(1)]$, we can observe that the derivatives at either endpoint of the Lorenz curve dictate the minimum and maximum values of the distribution. The Lorenz curve for a distribution with support on the entire real line (minimum income 0, maximum $\to \infty$) must then have a derivative of 0 at 0 and ∞ in the limit as it goes to 1. This is the case for the lognormal distribution, and - although it's not obvious from the plot above - we can see this to be the case if we zoom in a bit.
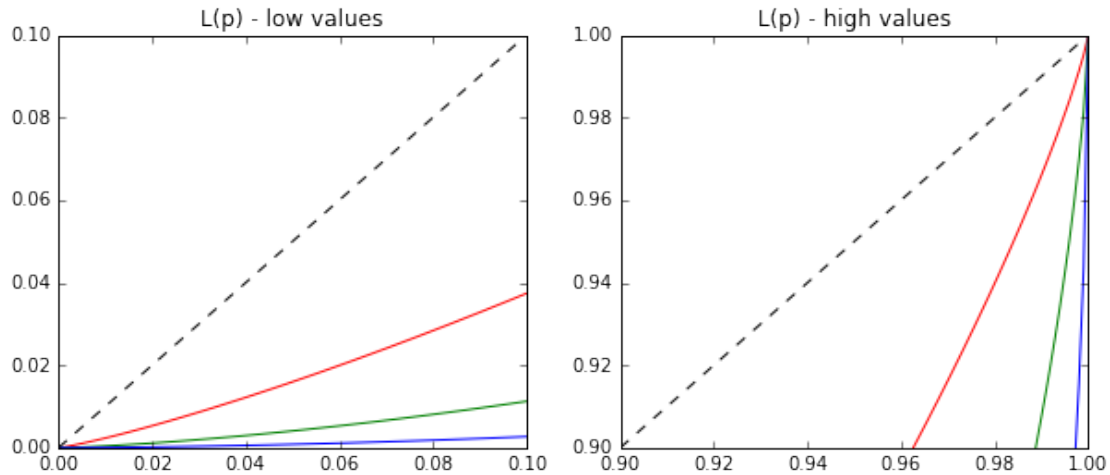
3.1.4 Constructing Lorenz curves from data

Recall from the introduction that a Lorenz curve has an obvious empirical analogue, which we constructed as part of our grouped data. We can review it here:

```
Out[6]:     $p$  cum_Y_p ($L$)
        0   0.1           0.03
        1   0.5           0.29
        2   0.7           0.48
        3   0.9           0.76
        4   1.0           1.00
```

It follows from the definition that $L(0) = 0$ and $L(1) = 1$. By construction the former is implicit here, while the latter is included in the table. (PovCalNet constructs Lorenz tables in the same way.) We can plot the points, and it should be clear that the curve is increasing and convex as required.
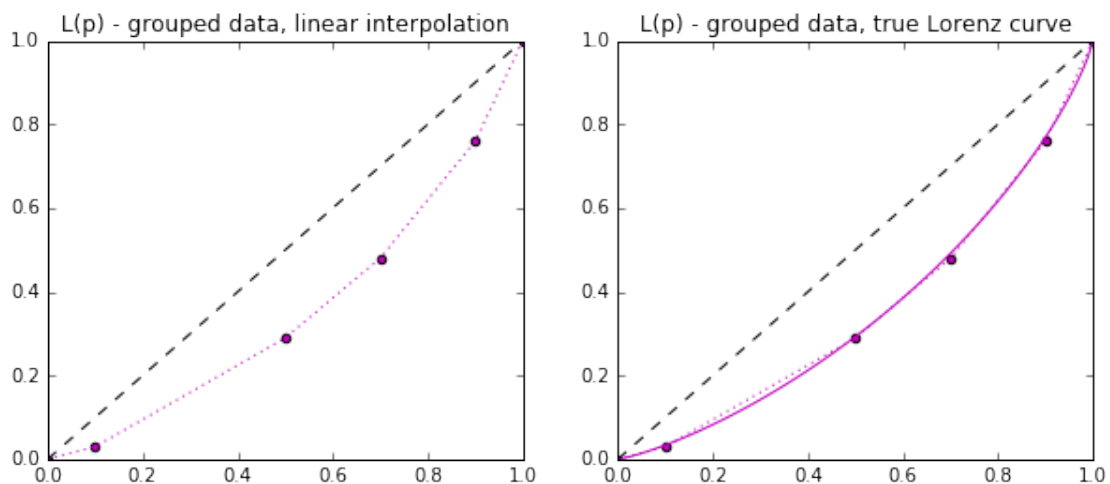
Since we only observe a finite number of Lorenz curve points, we do not normally know the 'true' shape of curve. On the left, we show a very natural interpolation - piecewise linear. On the right, we also show the 'true' Lorenz curve generated directly by the distribution from which we originally drew the sample.

The piecewise linear interpolation may be adequate for some purposes: for example, even with a small number of points, the Gini index so calculated will be quite accurate. However, comparing it with the curve at the right, we can see there are two sources of error:

- sampling error: some of the points do not lie exactly on the true line, because they are statistics of a finite sample
- interpolation error: in this case, the true line curves smoothly, and so sometimes deviates from the linear interpolation

There is an additional issue, which though technical is aesthetically quite important. Recall from the theory section that the quantile function (and hence its inverse, the c.d.f) depends on the derivative of the Lorenz curve. A piecewise linear function has a piecewise constant derivative, so the c.d.f. will be a discontinuous step function. Such a c.d.f. has no p.d.f, but instead has a probability mass function (p.m.f.) as it describes a discrete probability distribution. In this case, the p.m.f. will have atoms of mass equal to the entries in the Lorenz table, above.

This rendering of the Lorenz data is unlikely to reflect the true income distribution, which for a large population - while technically still discrete - would be better represented as a continuous distribution, with the atoms smoothed out over the support of the distribution.

This issue motivates the next chapter, which investigates various better methods of interpolating the grouped data. They are further illustrated in Appendix A.

Chapter 4

Fitting distributions to grouped data

The emphasis is on density functions, because a well-estimated density will usually result in good transformations (e.g. c.d.f, lorenz curve), whereas the reverse is often not true.

4.1 Common issues with Lorenz curve data

In practice, empirical Lorenz curve data derived from consumption or income surveys suffer from a number of issues that the theoretical treatment may ignore.

4.1.1 Negative and zero values

In consumption surveys, it is impossible - by construction - for a person to have negative consumption, and effectively impossible to have zero consumption (given minimum caloric requirements, etc.). In income surveys this is not the case, and many of the surveys including in PovCalNet indicate the presence of negative or zero incomes, in one of two ways:

- The Lorenz curve may be non-increasing. The theoretial condition that a Lorenz curve must be increasing and convex holds only for an income distribution which is non-negative and has no atoms. Negative incomes result in a curve which is decreasing initially, while atoms of probability (for instance at zero) result in sections which are flat. It is often the case, for example, that the first several $L(p)$ values are zero, which indicates that some non-zero proportion of the sample have zero incomes. In some cases, grouping may hide negative incomes (if the first group is a mix of negative and positive incomes), but the best fit will strongly suggest that negative incomes were present.
- More simply, the sample minimum may be reported as negative

  At present we do not attempt to deal well with negative or zero incomes. These usually represent around 1% of the sample, so we simply discard them and renormalize the distribution accordingly.

Example

Below, we show an example of data from PovCalNet with zero incomes (Venezuela, 2005 - apparently around 8% of the sample is zero, which is an extreme case). A pecularity of PovCalNet means that points are repeated in this scenario.

```
Survey minimum ($PPP/month) 0.0
```

```
Out[2]:        $p$     $L(p)$
      0   0.000000   0.000000
```

```
        1  0.083534  0.000000
        2  0.083534  0.000000
        3  0.083534  0.000000
        4  0.083534  0.000000
        5  0.083534  0.000000
        6  0.083534  0.000000
        7  0.083534  0.000000
        8  0.083534  0.000000
        9  0.090028  0.000113
```

And also, an example of data with negative incomes (Denmark, 2012 - only the decreasing portion of the curve represents negative incomes, so here, less than 2%).

```
Survey minimum ($PPP/month) -8251.09
```

```
Out[3]:          $p$      $L(p)$$
        0  0.000000   0.000000
        1  0.010533  -0.004321
        2  0.020433  -0.003299
        3  0.030760  -0.000712
        4  0.040859   0.002762
```

4.1.2 Tails, minimums & maximums

4.1.3 Smoothing

We can consider two dimensions to smoothing, one statistical and one aesthetic.

In the statistical (inferential) sense, smoothing is a trade-off between bias and variance, which arises because we are dealing with a sample rather than a complete population. If we fit a Lorenz curve that passes precisely through each point in the grouped data, we are likely to overfit to sampling noise - increasing the variance of the modelled distribution, resulting in artefacts. On the other hand, smoothing will regularize the distribution towards some restricted class of distributions, but this will inevitably bias the estimated distribution. In this frame, parametric distributions (say, a two-parameter lognormal) are just an extreme form of smoothing, extremely robust to sampling error but forcing the distribution into a form that is unlikely to be 'true'. Intuitively, more smoothing will be needed when the number of Lorenz points is high relative to the sample size (i.e. P/N is high), since this reduces the effective group size.

In an aesthetic sense, smoothing is important in emphasising salient aspects of the data. This is not a sampling issue, but a question of how best to communicate the population distribution. In reality, population income distributions are unlikely to share the properties of commonly used distributions like the lognormal or Pareto. They may exhibit clustering or spikes before certain thresholds (for example, $25,000 may be a much more common salary than $24,999, resulting in a spike). A true rendering of the distribution would reflect these features, but -- depending on the visualization -- a viewer may find them distracting, preferring to see the overall shape of the distribution. In particular, if a series of distributions is being compared over time, and these "systemic artefacts" move around (e.g because of currency conversion, or perhaps because thresholds themselves move), that can make it harder to understand the main changes.

(Another way of thinking of this is that even a population dataset is a random 'draw' from a hypothetical true, unknowable distribution, and that it is this distribution we would like to model. This would also argue for smoothing even of population data.)
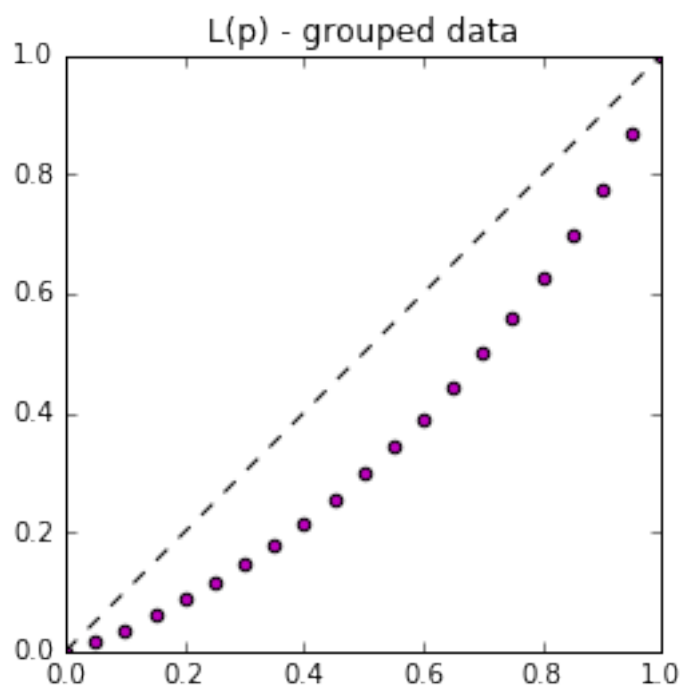
Ideally these dimensions would be separated

1. statistical smoothing would be handled during the fitting of the grouped data, to minimize some criterion (e.g. mean-square error)
2. aesthetic smoothing would be handled during visualization, dependent on the visualization

In practice we smooth once at the time of fitting. Some vizualisations (e.g. histograms) naturally impose an additional smoothing step (through bin size), but we are not rigorous about this.

Example

To understand the smoothing issue, it can help to see different degrees of smoothing. We first simulate a population of 1000 incomes from a lognormal distribution, then take a sample of 200, which we aggregate into 20 groups. The resulting Lorenz curve is plotted below.
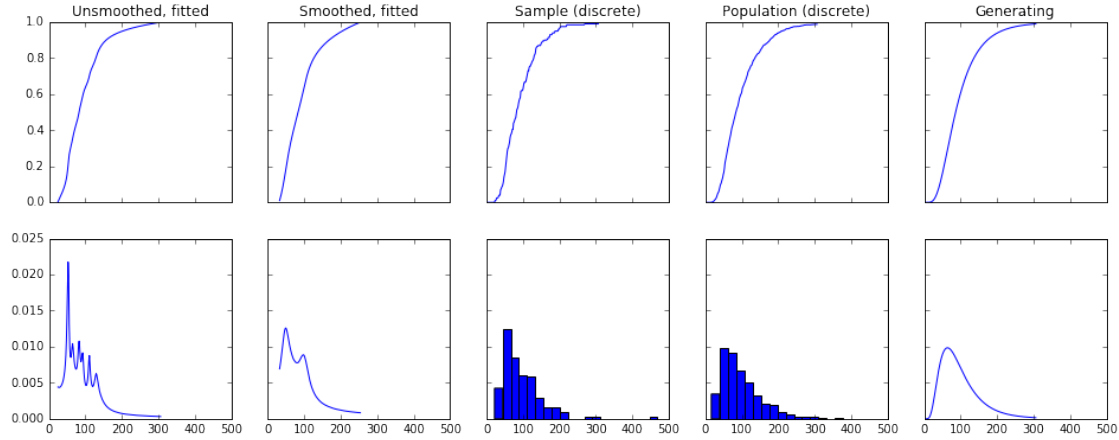

We then fit the Lorenz curve using a simple quintic spline (see below for explanation). Below, we show the c.d.f and p.d.f for (left to right):

1. The distribution inferred from the grouped data (no smoothing)
2. The distribution inferred from the grouped data (smoothing)
3. The (discrete) sample distribution from unit records
4. The (discrete) population distribution
5. The hypothetical/generating population distribution (ie. lognormal)

For 3 and 4 we show normalized histograms rather than the p.d.f (which does not exist) or the p.m.f. (which has too many points to be easily visualized ungrouped).

```
Out[7]: [<matplotlib.lines.Line2D at 0x112a85198>]
```

We can see easily that the unsmoothed fitted distribution, while faithful to the sample (compare the c.d.f.s), is overfitted, and results in a p.d.f which includes sampling artefacts. With smoothing, the fitted distribution is both closer to the population and generating distribution, and has a p.d.f which is much easier to interpret.

That said, the smoothed, fitted distribution still has some issues: it is too "peaky", and misses the left tail (which has not been captured in the sample). Issues like this lead us not to prefer the simple spline-interpolated Lorenz curve, although we consider it further below.

4.2 Fitting methods

Fitting income distributions from grouped data is a well-established problem, and many different approaches exist. We experimented with many of these, and will briefly survey them before describing the approach eventually taken.

In the survey, we demonstrate many of the methods on real data for four PovCalNet surveys:

```
Out[8]:       Country  Year  Coverage      Format
        0       Brazil  2014  National  Unit record
        1   Mozambique  2008  National  Unit record
        2    Indonesia  2014     Rural  Unit record
        3        China  2013     Urban      Grouped
```

4.2.1 Parametric

Parametric distributions often fit income data quite well, and easy to work with. They are used both in the academic literature (e.g. Lakner & Milanovic, 2013) and in other web visualizations (e.g. Gapminder, described above).
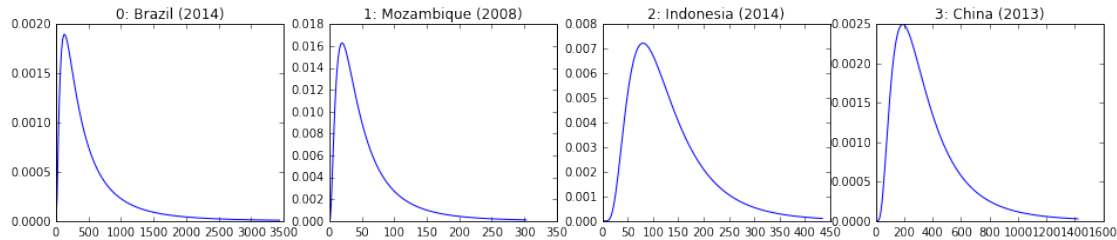
Parametric distributions are also used implicitly in PovCalNet itself, when unit record data are unavailable. In these cases, both a "general quadratic lorenz curve" and a "beta lorenz curve" are fit to the Lorenz points, with the superior fit being used to generate statistics.

Lognormal

The lognormal distribution (introduced in Chapter 2) is perhaps the easiest to fit from Lorenz curve data. It has two parameters, usually given as the mean μ and variance σ^2 of the normal distribution of log income. However, σ^2 can be calculated directly to the Gini index of the distribution, and μ can calculated from the mean of income. So from an estimate of the Gini index ($\hat{G}$) and of the mean ($\bar{Y}$), it is trivial to fit the distribution as follows:

$$\sigma = \sqrt{2}\Phi^{-1}\left(\frac{\hat{G}+1}{2}\right) \quad \mu = \log(\bar{Y}) - \frac{\sigma^2}{2}$$

The four example surveys have the following lognormal fits. Note that all the distributions have essentially the same shape, which is imposed by the parametric form. The income axis scaling is set so that the 99th percentile is at the right.

General Quadratic

The general quadratic method was introduced by ([Villasenor & Arnold, 1989](#)) for the specific purpose of fitting Lorenz curves. We will not the describe the details here, only note the method of fitting, which is by linear regression of the following equation in L and p of the Lorenz curve.

$$L(1-L) = a(p^2 - L) + bL(p-1) + c(p-L)$$

A feature of the general quadratic Lorenz is that it implies a potentially non-zero lower bound and finite upper bound for the distribution, given by:
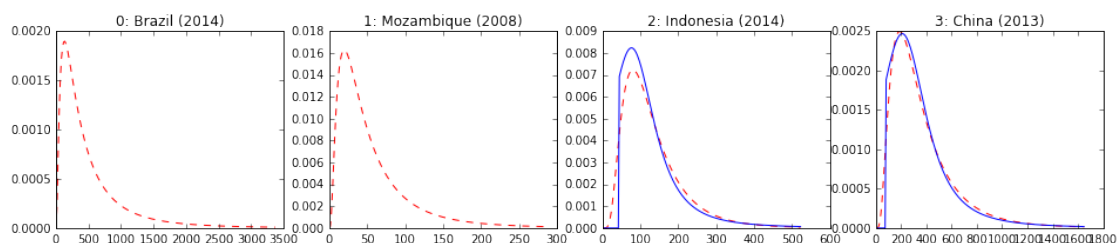
stuff

The density function is then:

more stuff

where K is a constant required so that the density integrates to 1 (between the lower and upper bounds), and which is calculated numerically.
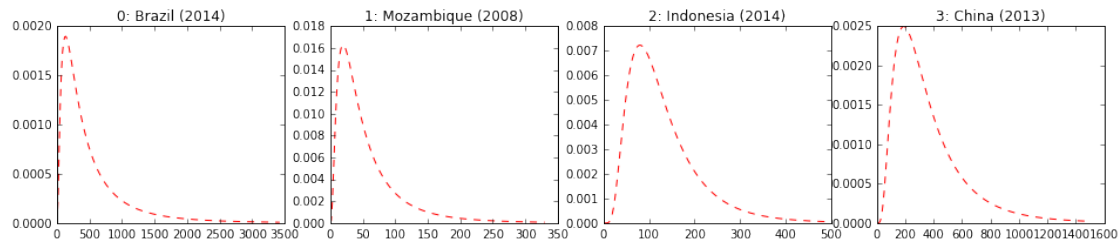
The four example surveys have the following general quadratic fits. The general shape is quite similar to the lognormal, but the non-zero lower bound potentially implies quite different results in the left tail.

```
../lib/income/distributions.py:338: RuntimeWarning: divide by zero encountered in true_divide
  return (1 + ((x - self.nu)/self.tau)**2)**(-3/2) / self.L1_normaliser * (x >= self.lower) * (x <= sel:
../lib/income/distributions.py:338: RuntimeWarning: invalid value encountered in multiply
  return (1 + ((x - self.nu)/self.tau)**2)**(-3/2) / self.L1_normaliser * (x >= self.lower) * (x <= sel:
```


17

Beta Lorenz

```
../lib/income/distributions.py:360: RuntimeWarning: invalid value encountered in double_scalars
  return p - self.theta * p ** self.gamma * (1 - p) ** self.delta
```


Post-adjustment

Shorrocks & Wan (2008) suggest a procedure for fitting a distribution to a grouped income data, by adjusting the output from a parametric fit (e.g. lognormal). Since the procedure creates a synthetic sample, rather than a distributional form (e.g. a p.d.f. or c.d.f), we did not closely consider it for this project. It is, however, worth noting (a) because it could be combined with another method (e.g. kernel density estimation) to estimate a distribution (this is especially true if a large sample is generated and (b) it is implemented in the Stata ungroup command of the popular DASP package and hence is quite widely used (e.g. by Lakner & Milanovic).

It would be valuable to include this method in a any future comparison.
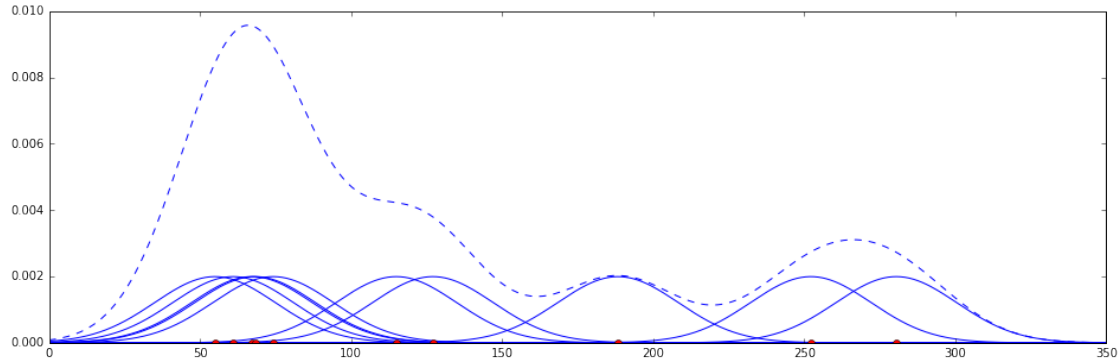
4.2.2 Kernel density

Kernel density estimation is the standard nonparametric approach to estimate a density function from a sample of observations. It solves the problem, noted above, that every sample technically has a discrete distribution, so that the derivative of an empirical distribution function --- perhaps the obvious choice of nonparametric density --- will always be ill-specified.

The basic idea is to represent each observation i not as an individual mass of probability at x_i, but as a unimodal 'hump' of mass, centered at x_i. With appropriate normalisation, the sum of these humps for all observations will result in a density function.

Smoothing, in kernel density estimation, is controlled by the 'bandwidth', the width of the hump. One great advantage of the method is that substantial intellectual effort has been dedicated to optimal choices of bandwidth smoothing, unlike many other methods.

We can demonstrate the kernel density method using a small sample from a lognormal distribution. In the plot below, the red dots represent observations, the solid blue curves represent the kernels ('humps') for each observation, and the dashed blue line represents the kernel density estimate, the aggregate of the kernels.

```
Out[12]: [<matplotlib.lines.Line2D at 0x1135e09b0>]
```

18

Based, as it is, on sample (unit record) data, it is not obvious how kernel density estimation should be adapted for use with grouped data. Despite this, one strand of the income distribution literature does this, particularly the papers of Sala-i-Martin (e.g. QJE 2006). The exactly method is unclear (synthetic observations would need to be generated) and some of the justifications seems spurious, and indeed in more recent work (NBER 2009) Sala-i-Martin appears to prefer parametric fits.

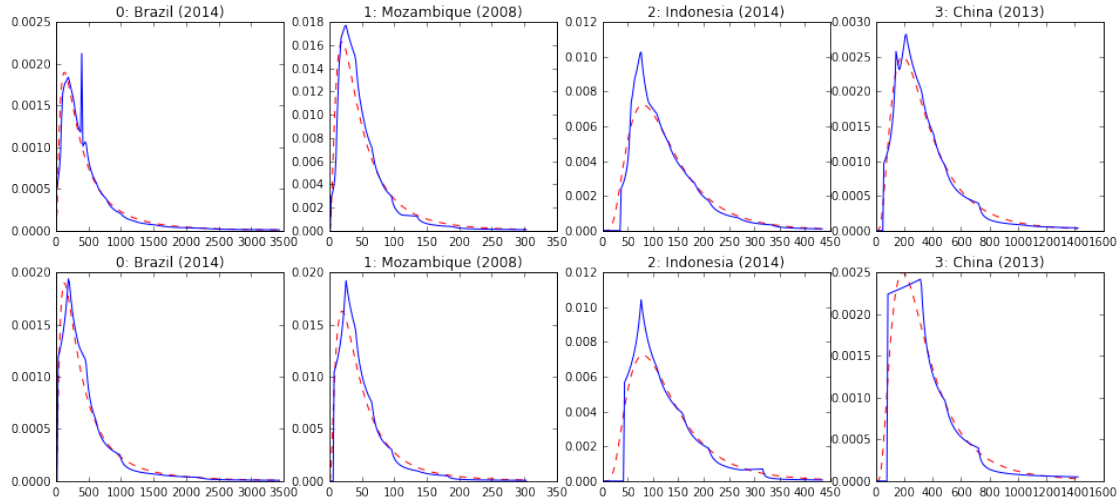We do not attempt to follow this approach.

4.2.3 Splines

Splines are piecewise polynomial functions in which continuity, and continuity of some derivatives, is enforced. They are widely used in engineering contexts as a general method of approximating smooth functions. They can be fit quickly by linear algebraic methods, and standard spline methods are widely implemented in numerical methods libraries. Conveniently, since splines are represented as segments of polynomials, they are closed under differentiation and integration. Unfortunately, they are not closed under inversion, so numerical methods are still required to move between the quantile function $Q(p)$ and the c.d.f $F(y)$ or p.d.f $f(y)$.

Linear and Cubic

Linear splines (splines of degree 1) are simply piecewise linear functions. At the end of Chapter 2, we outlined the reasons that fitting these to Lorenz curves is undesirable.

Cubic splines (splines of degree 3) are the most common in engineering applications. These provide a smooth Lorenz curve and a smooth, convincing distriution c.d.f. However, the p.d.f will be of degree 1 (piecewise linear), with sharp points and kinks.

See examples of cubic spline fits, below both without (top row) and with (bottom row) smoothing. Note that smoothing has little effect on the p.d.f (because it is acting at a higher level of integration) and, in particular, does not make the curves smooth in a geometric sense.
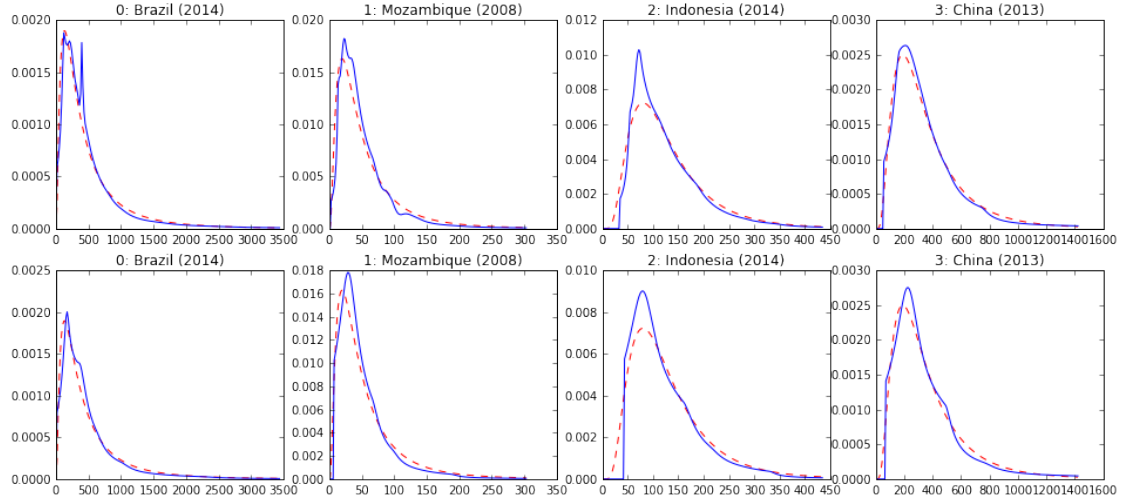
Although cubic splines are not well suited to this problem, as the first nonparametric method we've applied to the example distributions, we should pause briefly to examine the differences between these fits and the lognormal fits, which is again shown as a reference:

- For Brazil and Mozambique, the modal value appears to be somewhat higher (further right) than in the lognormal fit.
- For Brazil, there appears to be a potential second mode in the distribution. Although this diminishes with smoothing, the feature remains. (Comparison of multiple years of data, not shown here, suggests this is, in fact, a real feature of the income distribution, and not an artefact.)
- For Indonesia, the distribution seems to be substantially 'peakier' than the lognormal approximation would suggest.

We should not overstate these differences: they are probably not important for the calculation of most important statistics. However, in keeping with the project objective of showing distributional detail, they are interesting for visualization purposes, so our aim is to retain them rather than over-smooth them away.

Higher-order

Since the p.d.f. relies on twice differentiating the Lorenz cuve, it stands to reason that if we require a smooth curve for the p.d.f (i.e. a spline of order 2, where the spline nodes share both location and slope), we should use a spline of order 4 or above. Indeed this produces more attractive p.d.f.s than the cubic splines, as demonstrated below (again unsmoothed - top row, and smoothed - bottom row).

A number of important features are visible from this example. First, the smoothness property is visible in both rows. Second, the use of smoothing makes a noticeable difference this time, in eliminating sampling noise. Third, the distributions may have a non-zero lower bound (like the general quadratic parametric fit, but unlike the lognormal parametric fit we are using as a reference) -- moveover, this lower bound tends to increase as a side effect of smoothing.

Constrained

One major issue with using splines to model Lorenz curves is that splines are not, in general, constrained to obey the properties of Lorenz curves (i.e. end point values, non-decreasing monotonicity, convexity). Often this is not important, as these constraints will not be binding, and an unconstrained spline will have these properties anyway. But this is not always true.

Many approaches are discussed in the literature for constraining cubic splines (e.g. PCHIP). Such methods are much rarer for higher-order splines, and none are available in standard libraries.

One method that looks promising was recently described by Zhang, Wu and Li (2015, working paper). Following Ramsay (1998), the authors use a transformation of a spline that ensures the final output must be convex and increasing. If the spline itself is given by $m(t; \theta)$, with z the parameters, then this approach instead fits the curve:

$$ s(u; \theta) = \int_0^u \exp \left(\int_0^s m^2(t; \theta) \, dt \right) \, ds $$

The square of m must be non-negative so it's integral (and the exponential of its integral) must be non-decreasing, so the integral of that, s, must be (at least weakly) convex.

Some efficiency gain is given as the inner integral can be solved analytically, however the outer integral still requires a numerical solution, so solving for the parameters (which requires iteration) is slow compared with an unconstrained spline. In our case, we found it too slow.

4.2.4   Semi-parametric

We conclude this review with two methods with combine aspects of parametric estimation with non-parametric estimation. First is the method actually used, which we call *parametric-Lorenz-transformed spline* fitting. The second is a recently published method used by the WID project, which shares some similarities (albeit applicable to a slightly different context).

Lorenz-transformed spline

This method is designed to address some of the issues with straightforward high-order spline fitting on the Lorenz curve. The key idea is that the overall shape of the Lorenz curve can be represented using a parametric distribution, while the deviations from that parametric form can be captured using a spline.

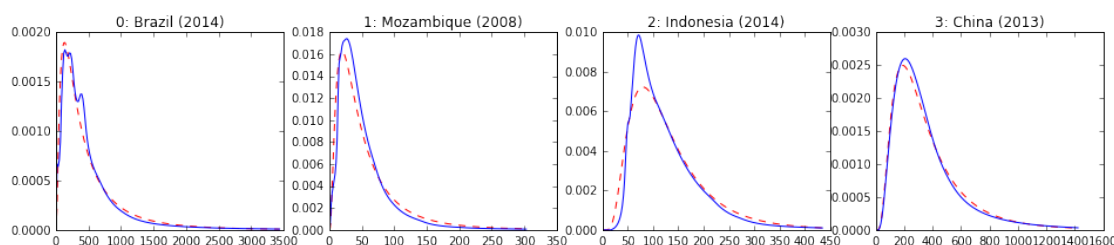The steps we perform are as follows. Given the Lorenz points $\{p_i, L_i\}$

1. Calculate the Gini index and using that and the known mean, fit a lognormal distribution with Lorenz curve $L_{lognormal}$

2. Transform the points {L_i} using the inverse Lorenz curve, $p_i^* = L_{lognormal}^{-1}(L_i)$

3. Fit the curve $\{p_i, p_i^*\}$ using a smoothed high-order spline. If original Lorenz points came exactly from a lognormal distribution, then this curve should be a 45 degree line. More realistically, when the distribution is close to lognormal in shape, there will be minor deviations from a 45 degree line, which the spline will fit.

One major advantage of this is that in the tails, the (exponential) lognormal Lorenz curve will tend to dominate the (polynomial) spline, so the tails will be effectively determined by the parametric choice rather than the spline. The resulting distribution will, for instance have support on $[0, \infty)$, which is convenient for manipulation.

Additionally, the lorenz curve of the lognormal is composed of the Φ and Φ^{-1} functions, which are available in efficient implementations in most stats/probability libraries. Further, the first derivative of the lognormal Lorenz-transformed spline can be calculated analytically, which improves computation time for $Q(p)$ (numerical inversion is still required for $F(y)$).

(Of course, the same technique could, in principle, be applied to a different parametric Lorenz curve).

For our example distributions, the method produces the following p.d.f.s:

Note that, in common with the parametric lognormal fits, these densities descend smoothly to zero, and asymptote to infinity. At the same time, they preserve some of the detailed features of the distributions found in the spline fits, above (for example, the bulge in the Brazil density, and the peak of the Indonesia distribution).

> Above we have used a fixed smoothing parameter. In practice, for the present visualisation, we select the smoothing parameter based on some heuristics: with aim for the minimum smoothing that eliminates spikes in the distribution and results in no more than X local maxima.

Postscript: generalized Pareto curves

During the course of this project, the WID project published Blanchet, Fournier and Piketty (March 2017) (based largely on Fournier's 2015 master's thesis). This method of fitting grouped income data requires access to the group thresholds, which are not available in PovCalNet output, so we are unable to test it. It does, however, share conceptual similarities with our preferred Lorenz-transformed spline method, in particular the spline fitting of a transformation of the grouped data. For this reason we briefly outline it now.

The method takes its inspiration from the Pareto distribution, commonly used to model income distribution, which has the c.d.f

$$F(y) = 1 - \left(\frac{y}{y_{min}}\right)^{\alpha}, \ y > y_{min}$$

One property of this distribution is that the mean of incomes above any threshold, divided by the threshold, is constant. That is, for all y,

$$\frac{E(Y|Y > y)}{y} = \frac{\alpha}{\alpha - 1} = b$$

The authors generalize this property to non-Pareto distributions by first allowing $b(y)$ to vary in y, and then by reparameterising b using $y = Q(p)$ to arrive at a function on $[0, 1]$, $b(p) = b(Q(p))$.

From grouped data with thresholds, one can then calculate - and plot - the sample analogue of this function, at the observed group thresholds. From there the authors use high-order spline interpolation to fit the entire function, which can then be used to reconstruct the density.

Although this method cannot be used to solve our problem, it could be compared with the Lorenz-transformed spline method (by simply discarding the threshold data), which would be a wortwhile exercise.

Chapter 5

Canonical representation of distributions

The objective of this project is to display, and allow the user to manipulate, various transformations and statistics of income distributions, for example:

1. Showing any of the p.d.f, the c.d.f, or the Lorenz curve, etc
2. Calculating Gini coefficient, extreme poverty incidence, etc
3. Combining distributions (e.g. for all EU countries)

In order to permit this, we required a standard representation of an income distribution which is compact in size (since hundreds of these must be downloaded to initialize the app) and which permits fast transformation such as the above (since we cannot possibly pre-calculate all the possible combinations).

This motivates our definition of a 'canonical' representation of an income distribution, based on simple linear splines with an optimized set of knots.

5.1 Rejected representations

Several obvious representations were rejected - we briefly outline the reasons here:

1. **Common parametric forms.** These have the advantage of being extremely space efficient (e.g. two real numbers to parameterize a lognormal distribution) and relatively computationally efficient (since for common parametric forms, transformations are mostly available analytically, and are composed of standard function). However, as noted above, we consider these two constraining, imposing strong assumptions on the shape of the distribution.

2. **Lorenz curve spline as estimated.** The end result of the fitting process described in the previous chapter is essentially a fourth order spline with knots at the original income groups, resulting in a representation of at most several hundred real numbers for a given income distribution. However, calculating the points of the density function requires numerical inversion and differentiation, which are both relatively slow iterative processes, so we reject this option.

3. **Gridded, precomputed functions** We could precompute the Lorenz curve, c.d.f and p.d.f over a grid of values (say, 1000, to achieve a smooth appearance), and use this as our representation. This would, of course, be fast, but then each distribution representation would require ~3000 real numbers, or around 12 KB per distribution. As there are around 1400 surveys represented in PovCalNet, this would result in an initial download of 16MB. While not prohibitive, this is still quite large on, e.g. mobile devices.

5.2 Adaptive linear spline representation

The canonical respresentation we adopted was based on a spline representation of the density function, with a fixed number of knots chosen automatically by algorithm to minimise the error. The steps are as
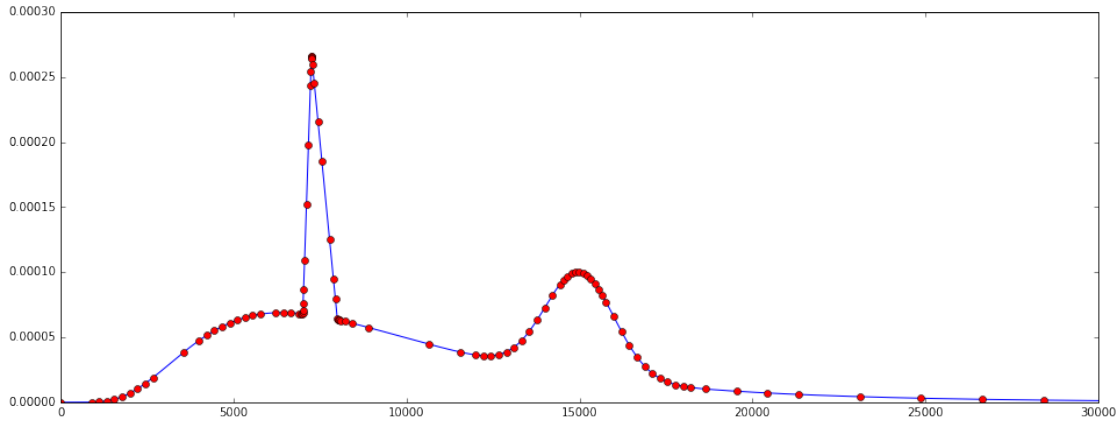
follows:

1. Using numerical transformations of the estimated Lorenz curve, generate a dense uniform grid of points of the density, on the interval $[0, Q(p_{max})]$ (we use $p_{max} = 0.99$ presently).

2. Fit a new smoothed spline, of order 1, to this grid of points, varing the smoothing parameter until the resultant spline has around 100 knots. These knots, $\{y_k, f(y_k)\}$ represent the distribution. (The number 100 is somewhat arbitrary, but was adequate to produce visually smooth curves in all the cases we examined.)

Although a higher-order spline would improve precision of the representation, there are advantages to representing the p.d.f in this piecewise linear function - in particular, it allows efficient calculation of the c.d.f, quantile function and Lorenz curve. Moreover, using (for example) a quadratic (order 2) spline representation would double the size of the representation from 100 to 200 points. Although we haven't tested it, we presume the representation error of a 200 point adaptive linear spline is less than a 100 point quadratic spline.

Below, we show an example of a ~100 point adaptive linear spline fit to an artificially complex mixture distribution. The blue line is the true density, while the red dots are the points selected to represent the density with linear segments. As it demonstrates, even a small number of points produces an adequate fit, because the points are placed in areas of greatest deviation from linearity.
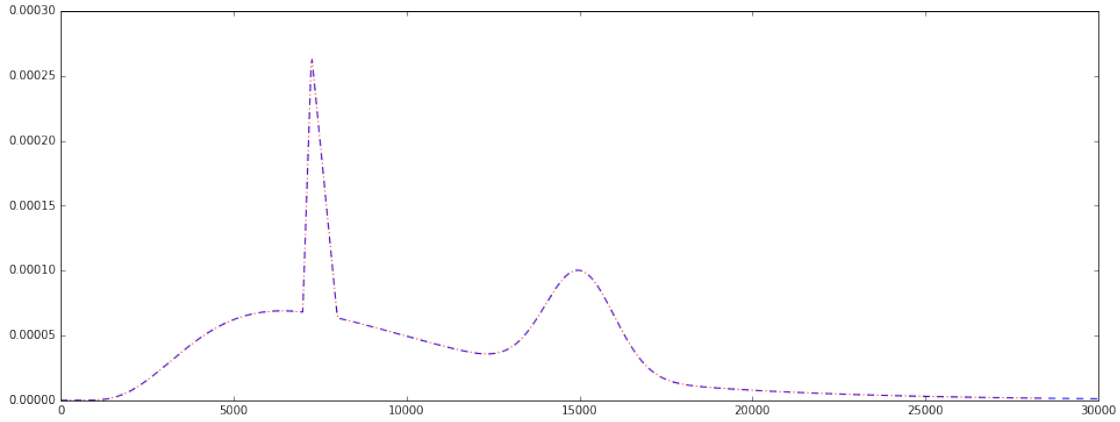
`99 knots`

5.3 Transformations

With a density represented by y_k, f_k for $k = 1 \ldots 100$, we can compute points on the major curves efficiently. In the app, these are implemented on the client (browser) side in Javascript.

5.3.1 Density

Simple linear interpolation. For y with $y_m <= y < y_{m+1}$,

$$f(y) = f_m + \frac{y - y_m}{y_{m+1} - y_m} \cdot (f_{m+1} - f_m)$$

25

We can reconstruct the density curve from above using the 100 point representation. It is plotted below as red dots. As can be seen, the error compared with the true density (blue dashes) is unnoticeable.
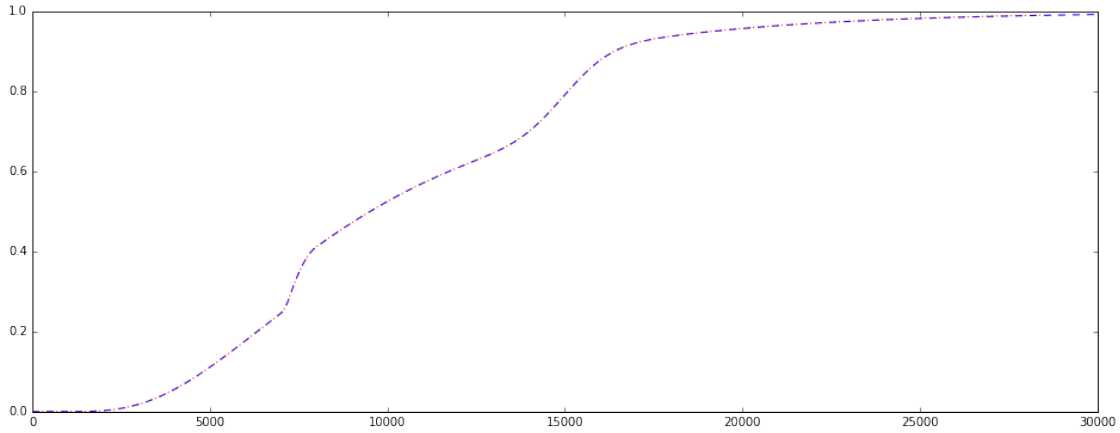

5.3.2 Distribution

Integration to quadratic, then cumulation. For y with $y_m <= y < y_{m+1}$,

$$F(y) = \left(\Sigma_{k=0}^{m} \int_{y_k}^{y_{k+1}} f_k + \frac{t - y_k}{y_{k+1} - y_k} \cdot (f_{k+1} - f_k)\, dt\right) + \left(\int_{y_m}^{y} f_m + \frac{t - y_m}{y_{m+1} - y_m} \cdot (f_{m+1} - f_m)\, dt\right)$$

which can be solved straightforwardly. The partial sums of the first term are pre-computed on first load for all k to improve speed.

5.3.3 Quantile function and Lorenz curve

The quantile function uses the same precomputed array of c.d.f values, doing a reverse lookup to find the relevant segment. Then, the distribution function is inverted analytically to produce the precise value.

The Lorenz curve can then be computed by analytically integrating the quantile curve. Since it is normalised by the mean, that too must be computed numerically (done once at load time).

5.3.4 Other statistics

Most other statistics that we present are easily calculated from one of the above functions: that includes median, extreme poverty headcount, Gini index, histograms.

5.4 Aggregations and interpolations

Two types of aggregations are used in the app, and we take a different approach to each:

1. For aggregating two distributions (e.g combining urban plus rural to get a national distribution, or interpolating between two years), we take the union of both sets of y_k, then generate new p.d.f points by a simple weighted linear combination - resulting in an new aggregate representation with ~200 points.

2. For aggregating many distributions (e.g. to generated the aggregate distribution for a region or other set of countries) we generate a new uniform grid with 1000 points, since the above approach would result in too many points. It's possible the above method could be adapted by first thinning the aggregate distribution back to 100 points before adding each subsequent distribution, but we haven't investigated this.

In either case, transformations of the aggregate distribution now follow as above.

Chapter 6

Data extension

This section is subject to change. We have used a set of simple rules for interpolation, extrapolation and harmonization in order to have a sufficiently large set of distributions available to test out visualizations in the prototype. **We have erred heavily on the side of generating longer time series, rather than accurate data. Once we refine the visualizations we should probably revert to a more conservative set of rules.** At a minimum, the app should very clearly distinguish between survey years and imputed years.

The set of Lorenz tables extracted from PovCalNet represents only actual survey years. Moreover, in some cases surveys are only representative of a subpopulation (e.g. urban residents). Here we briefly summarise the rules we apply to grow this set of around 1,400 distributions to cover a larger period of time. These steps follow in the sequence below.

6.1 Interpolation

When two distributions exist for the same measure (consumption or income) and the same population (same country, and either urban, rural or national), we do a simple time-weighted linear interpolation between the two densities to generate intervening years. That is, for year t without a survey, but bounded by two survey years such that $t_a < t < t_b$, we set:

$$f_t(y) = \frac{t - t_a}{t_b - t_a} \cdot f_{t_b}(y) + \left(1 - \frac{t - t_a}{t_b - t_a}\right) \cdot f_{t_a}(y)$$

Alternatives: We could weight the linear interpolation by some macroeconomic variable that is available in all time periods (e.g. household final consumption expenditure per capita or GDP per capita), as we do for extrapolation. This would have more impact when the gaps between surveys are larger (ie. more than a year or two).

6.2 Extrapolation

For years beyond the range of available surveys, we extrapolate. First, we extrapolate based on household final consumption expenditure (HFCE) per capita, to the extent that it is available. Then, we extrapolate based on GDP per capita. All extrapolation merely scales the distribution about zero, which assumes distributional neutral growth (or shrinkage) of all incomes, and will not, for example, change the Gini index. For example, to extrapolate back to year t where $t < t_a$, the earliest survey, we use:

$$f_t(y) = f_{t_a}\left(\frac{M_{t_a}}{M_t} \cdot y\right)$$

where both M_t and M_{t_a} are either HFCE per capita or GDP per capita, depending on availability.

Alternatives: We could use other macro variables for extrapolation. We could use an external Gini index series to attempt to extrapolate shape changes in the distribution over time as well.

6.3 Combining rural and urban

When both rural and urban subpopulation distributions are present, we build a national distribution using a population-weighted average of the two densities. That is, for year t:

$$f_{t,national}(y) = P_{t,urban} f_{t,urban}(y) + P_{t,rural} f_{t,rural}(y)$$

where P is the relevant population.

Where only one of rural or urban is available we assume this distribution applies to the entire population (recognizing that this is an obviously flawed assumption).

Alternatives: The main alternative here would be to simply ignore years where only urban or only rural distributions are available.

6.4 (Not) Harmonizing consumption and income

Like most other work in this area, we make no attempt to deal with the inconsistency of some countries using income surveys while others use consumption surveys. This would take effort substantially beyond the scope of this project, although it would no doubt be highly worthwhile.

When a country has both a consumption and an income series, we use the consumption series only and the income series is ignored.

Alternatives: We could use whichever series was longer, income or consumption (although since this stage happens after extrapolation, the expanded series of both will usually be the same length.

Chapter 7

Visualizing distributions

For offline reading, this chapter briefly summarizes the visualizations that are currently available, including a brief summary of major design choices.

Examples are shown for eight countries of the Communauté Financière Africaine (CFA) currency union: Benin, Burkina Faso, Guinea-Bissau, Côte d'Ivoire, Mali, Niger, Senegal and Togo, for the year 2000 (which involves extrapolated distributions in some cases). The legend as it appears in the app is reproduced in Figure 7.1

At present when multiple countries are selected, their distributions are aggregated (resulting in stacked visualizations or a single aggregate), rather than compared side-by-side or in overlay. In future it would be desirable to have both options. Additionally, all distributions are weighted by population (necessary for meaningful visual aggregation), which should be optional in future. Finally, as with the other parts of this project, this is a prototype, "minimum viable product": features like label axes, hover text and so on should be added in future.

The chapter concludes with a discussion of other visualizations that would be valuable to include in further work.

7.1 Density (p.d.f.)

The density plot (Figure 7.2) is standard: weighted by population, and stacked and colored by country. The income axis limits are determined so that the 95th percentile of all countries included is displayed.

7.1.1 Log axis

In the log income (x) axis version of the plot (Figure 7.3), the limits are fixed at 1 and 10,000 for all countries, in order to provide a globally comparable visualization. The density value is scaled by the x axis value in order to create areas that reflect total population of each country. As a consequence, the y axis unit is no longer constant along the x axis, and loses meaning, so the y axis is not labelled.

7.2 Distribution (c.d.f.)

The distribution plot (Figure 7.4) is the counterpart of the density plot. It is generally uninteresting.
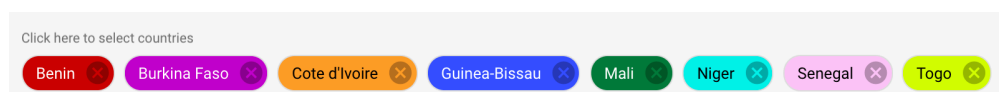

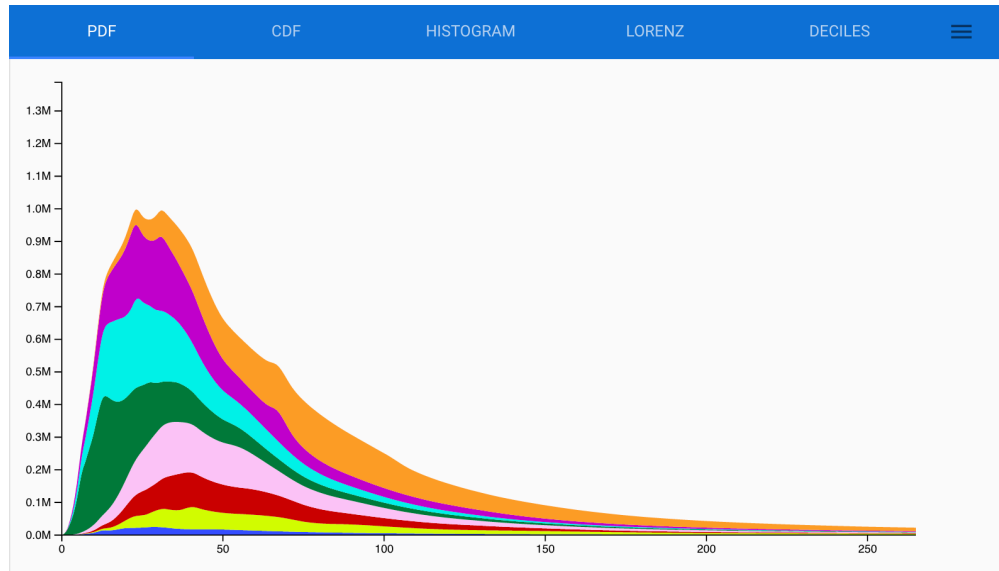**Figure 7.1:** *Screenshot of the legend for the selection of CFA countries*

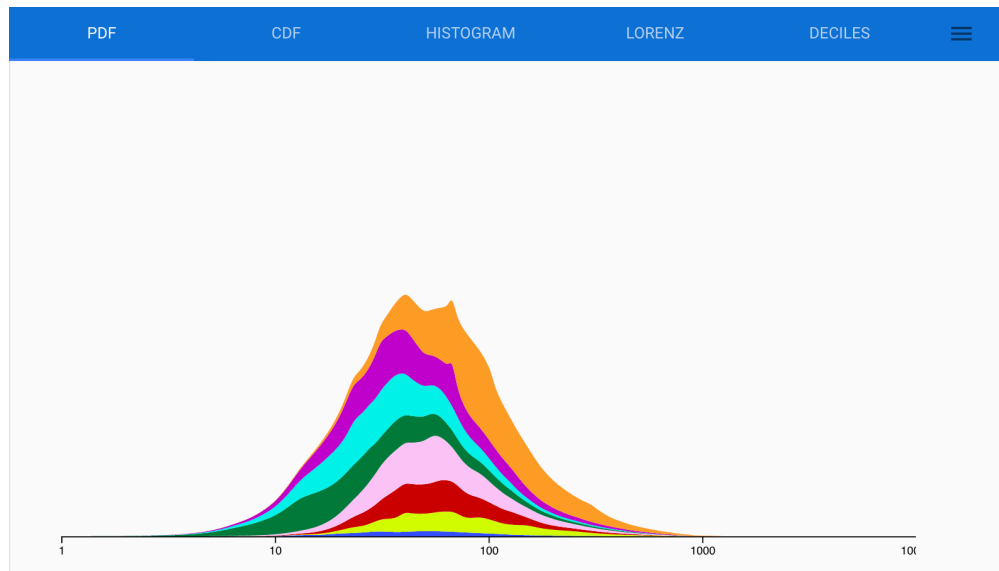*Figure 7.2: Screenshot of the PDF plot, ordinary linear income axis*

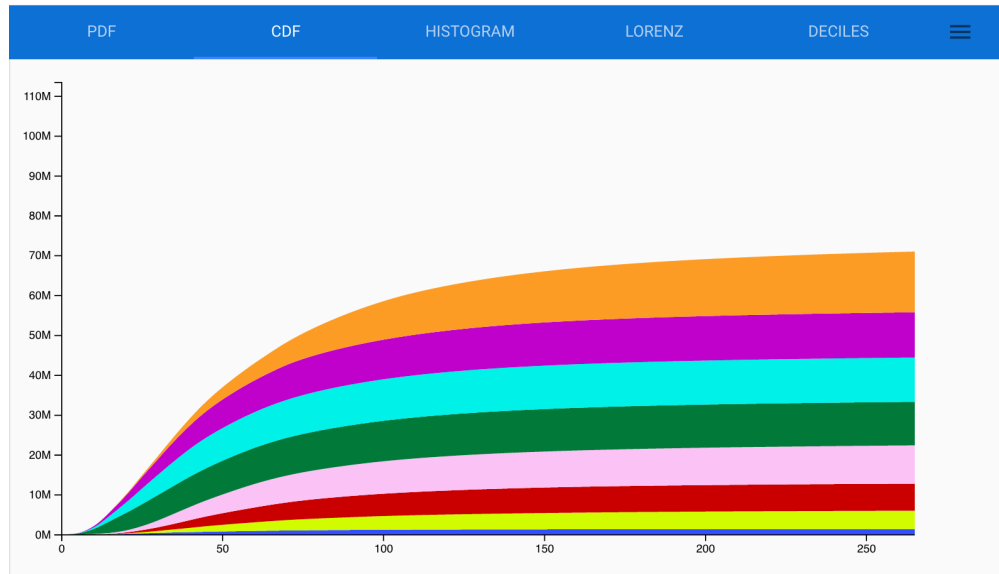*Figure 7.3: Screenshot of the PDF plot, logarithmic income axis*

Figure 7.4: *Screenshot of the CDF plot, ordinary linear income axis*

7.2.1 Log axis

The log income axis distribution plot (Figure 7.5) is the counterpart of the log income axis density plot. It is generally uninteresting. Note that for the distribution plot, the y axis values are not rescaled, as area is not interpretable on this chart. Although areas are shaded, this is to indicate stacking rather than draw attention to area as such.

7.3 Histogram

Histograms provide a similar perspective to densities, with a couple of advantages. First, with relatively few bins, they provide a more-smoothed view of the distribution than densities, eliminating potentially distracting spikes. Second, whereas the density y axis is only meaningful under integration (which a viewer will generally not do in their head), the y axis of a histogram shows actual totals for each column.

In the example plot (Figure 7.6), for example, this allows a viewer to roughly estimate that around 4 million people in these countries live on less than ~$PPP 13 per month, and that most of those live in Mali (looking at the first column). This sort of quick calculation is not easy on a density plot.

The histogram in the app is designed to be visually consistent (ie. in scale) with the density. Hence, the same 0 to maximum 95th percentile limits are used. This range is divided into 20 equal-width bins, with the final bin actually incorporating the entire right tail of the distribution, even beyond P95 (as is common, or even conventional, in income histograms).

7.3.1 Log axis

The log income axis histogram (Figure 7.7) involves the most design choices, since there is not an obvious way to depict this. Two obvious approaches were rejected:

- binning the data in non-log income space, which would result in a transformed histogram but without any benefit beyond the regular histogram
- binning the data in equal-wide intervals in log income space, which would result in uninterpretable bin boundaries

Instead, we constructed a set of bins that are approximately equal-width in log income space, but are adjusted to fall on interpretable boundaries. These boundaries are still essentially, with the exception of $58
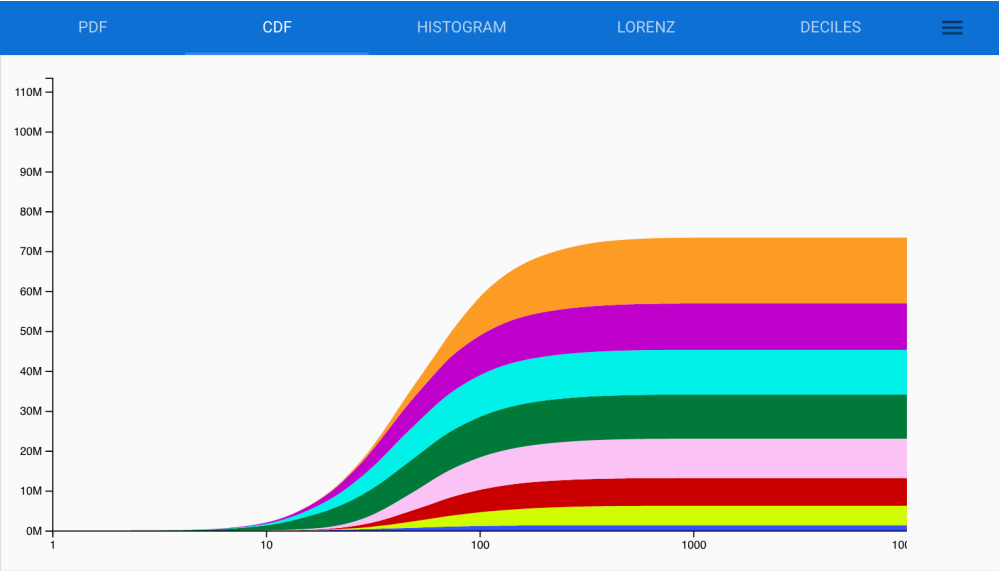
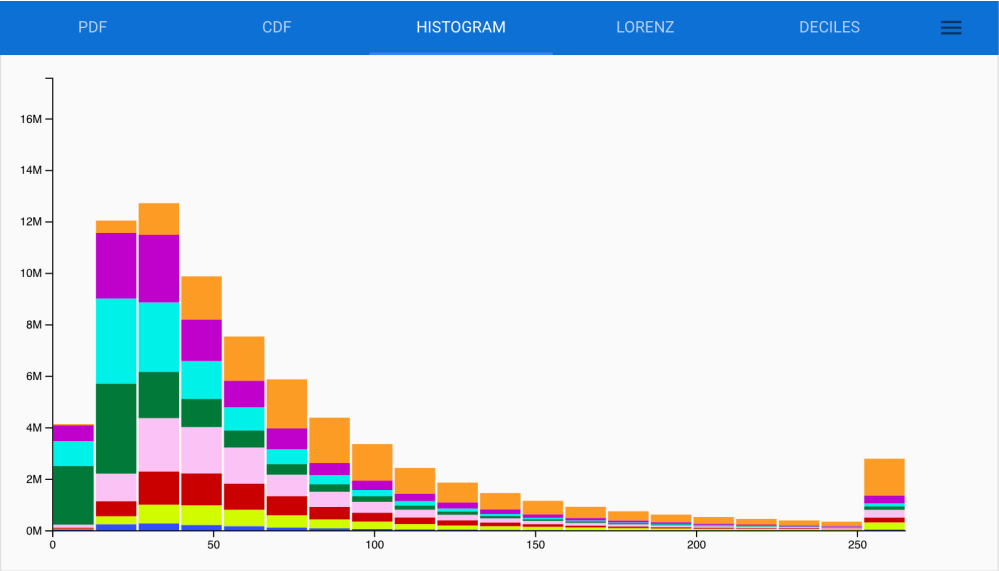**Figure 7.5:** *Screenshot of the CDF plot, logarithmic income axis*

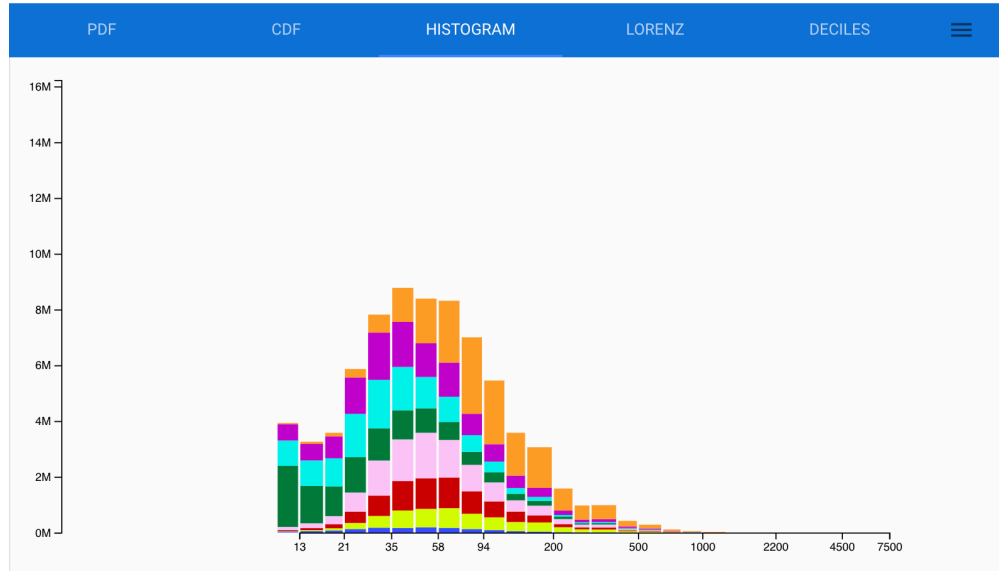**Figure 7.6:** *Screenshot of the histogram plot, ordinary linear income axis*

Figure 7.7: Screenshot of the histogram plot, logarithmic income axis

PPP, which is the threshold of extreme poverty ($1.90 per day), and $94 PPP, which equate to the higher $3.10 per day concept.

Here the column widths indicate the bin boundaries, and so vary slightly over the bins. They should not, however, be interpreted in an areal sense. For this reason, the lowest bin (which in reality extends down to 0) is shown with a false width, so as not to overwhelm. With the exception of this left-most bin, the chart is scaled to be visually comparable with the log density.

7.4   Lorenz curve

The Lorenz curves are shown straightforwardly (Figure 7.8). Since Lorenz curves cannot be decomposed, the individual country Lorenz curves are shown, colored by country, as well as the aggregate Lorenz curve for the selection, colored black.

7.5   Deciles

The deciles chart (Figure 7.9) shows average income per income decile in the population. As with the Lorenz curve, this cannot be decomposed by country, and only the aggregate is shown. This is the chart that most clearly demonstrates error in the right tail, since the highest decile is usually underestimated compared with other sources. There are at least two reasons for this:

- In the modelling stage, we only represented up to the 99th percentile, so the final percentile with the highest incomes is missing. This could easily be fixed by increasing the range to, say, the 99.99th percentile.

- However, even then, household survey data of the kind in PovCalNet is well known to fail to capture the highest income earners (this being the reason the WID project prefers taxation records as a source of data).

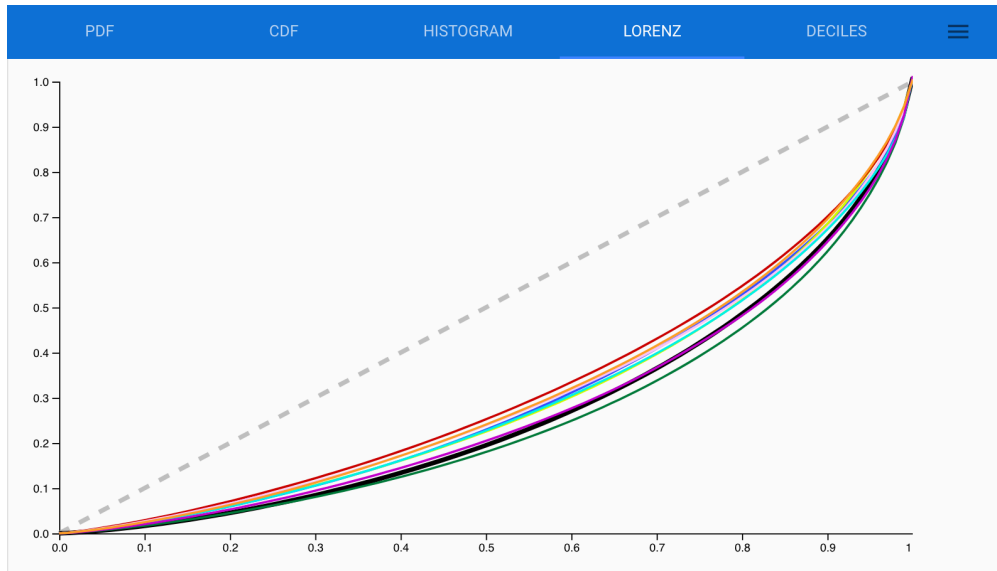Despite this underestimatation, this chart is still very effective in illustrating income disparities.

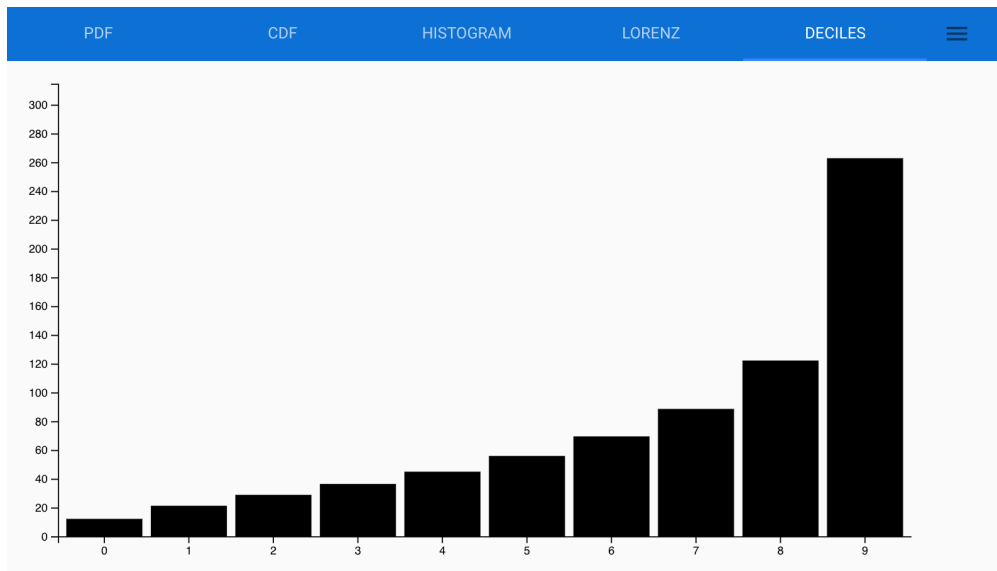**Figure 7.8:** *Screenshot of the Lorenz curve plot*

Figure 7.9: *Screenshot of the deciles plot*

7.6 Future chart types

A number of other chart types were considered during the course of this project, and should be tested for inclusion in future. In particular, all the charts currently depict snapshot of a point in time - although the user can interactively change the time point, there is no way to directly compare distribution characteristics at different time points. Options for this could include:

- growth incidence charts, which compare income growth at different quantiles of the income distribution, between a base year and a final year
- time series plots of income shares or means (e.g. top 10% share, bottom 40% share)