UrbanPy: A Library to Download, Process and Visualize High Resolution Urban Data to Support Transportation and Urban Planning Decisions

Andres Regal, Claudio Ortega, Antonio Vazquez Brust, Michelle Rodriguez, and Patricio Zambrano-Barragan

Abstract Research demonstrates that the large-scale access and analysis to data can help cities plan for more inclusive and efficient growth. However, policy makers and researchers around the world still lack sufficient access to granular and dynamic sources of data – a challenge that has come to prominence as cities attempt to respond to the local impacts of the COVID-19 pandemic. To address this challenge, this paper presents UrbanPy, a new open-source library that makes the automated collection, processing, and visualization of spatial urban data simple and consistent for cities. UrbanPy, first developed as a tool to help Latin American cities design rapid responses to COVID-19, presents six innovative capabilities for researchers and practitioners focused on data collection, processing and visualization. To illustrate the potential applications of UrbanPy, this paper presents a case study from Lima, Peru, where the library helped municipal leaders with their COVID-19 response by identifying vulnerable populations, creating food accessibility metrics, and optimizing the location of food supply facilities.

Keywords Urban data · Transportation · Urban planning decisions

1 Introduction

Cities today are home to close to 60% of the world's population and are collectively responsible for an increasingly larger share of economic output (UN-Habitat 2016). Given current trends, by the end of the XXI century 85% of people will live in cities, primarily in developing countries in Africa, Asia, and Latin America (OECD.

A. Regal (✉) · C. Ortega · M. Rodriguez
Universidad del Pacifico, Lima, Peru
e-mail: a.regalludowieg@up.edu.pe; c.ortegaariza@up.edu.pe; rodrigezm@up.edu.pe

A. Vazquez Brust · P. Zambrano-Barragan
Inter-American Development Bank, Washington DC, USA
e-mail: pzambrano@iadb.org

J. Vargas Florez et al. (eds.), *Production and Operations Management*, Springer Proceedings in Mathematics & Statistics 391,
https://doi.org/10.1007/978-3-031-06862-1_34

Publishing 2015). Rapid urban growth in LAC is the result of agglomeration benefits: people want to live in places that offer opportunities for a better life, while firms locate where demand and talent lie, thus providing conditions for human development through the effective delivery of basic services and the establishment of social and political organizations to solve difficult human problems (Glaeser 1999, Duranton and Puga 2014). However, unplanned urban growth can also present considerable challenges, particularly unequal access to quality infrastructure services and negative environmental impacts amidst sprawling urban expansion. Overall, these characteristics make cities "*the example par excellence of complex systems: emergent, far from equilibrium, requiring enormous energies to maintain themselves, displaying patterns of inequality spawned through agglomeration and intense competition for space, and saturated flow systems that use capacity in what appear to be barely sustainable but paradoxically resilient networks*" (Batty 2008).

While rapid urban growth, with all its challenges and opportunities, has defined global development since at least the mid-twentieth century, there is another, equally transformational dual phenomenon: digitalization and rapid technological change. Consider that, between 2016 to 2017, 300 million people started using the internet (DOMO 2020), increasingly via mobile devices (GSM 2018). Much of this growth has been driven by developing regions, including Latin America, where over the last three decades, the proportion of people using the internet jumped from 0% to 56% (World Bank 2020). Yet, much like urbanization, digitalization and technological change present challenges, ranging from privacy and security breaches, to overreliance on potentially biased data processing tools, and to persistent conditions of inequality in terms of access. The question is whether both urban growth and digitalization can benefit citizens.

Altogether, the possibilities new technologies present to urban development are concrete and likely to grow substantially. Recent reports show that the application of existing data-driven solutions could improve quality of life indicators by 10–30% in areas such as mobility, security, healthcare, energy, water, waste, civic engagement, economic development and housing (Woetzel et al. 2018). Dynamic analysis of urban data is thus essential to the proper management of cities and, when committed to principles of inclusion and open data, can improve citizen participation and policy outcomes (Townsend and Zambrano-Barragan 2020).

How can growing cities enable data-driven governance? Contemporary urban systems already generate tremendous amounts of spatial information – through mobile and transportation GPS trackers, geolocated data on land use and cadaster, and broader internet of things (IOT) – which can, in turn provide critical insights to improve living conditions. At present, there are proprietary and open-source tools for technical spatial analysis (ArcGIS and QGIS, respectively), as well as more user-focused platforms for navigation and location services in the city (Google Maps, and the crowdsourced open platform, OpenStreetMap). However, from the perspective of policy makers and city leaders, proprietary tools can be prohibitive in terms of cost and technical capacity, while open source solutions lack spatial resolution and functionality.

In this context, several efforts have been made to produce an open-source set of processing and visualitazion tools. Overpass API, OSMnx (Boeing 2017) and the Nominatim API aim to facilitate the gathering and processing of road network data, while Geopandas (GeoPandas developers 2020) abstracts spatial operations with points, polylines, and polygons. However, the workflow for managing several spatial data sets with different levels of aggregation, with disparate coordinate systems and sources, becomes increasingly more difficult as more layers of data are added to the analysis.

To address these limitations and to support data-driven urban governance, this paper presents UrbanPy, a library to download, process and visualize high resolution urban data. This open source tool facilitates best-available open data collection (using sources like OpenStreetMap and Humanitarian Data Exchange), while supporting standard spatial aggregation of data sets using regular polygons (i.e. grids or hexagons), as well as fast processing of spatial aggregations into these regular polygons. Ultimately, UrbanPy enables the visualization of these disparate data sources – and the combined layers and statistics built from them – thus strengthening decision support systems for key policy actors such as municipal authorities. Working with shortest path calculation, population density layers (grouped by age brackets), and Points of Interest (POI) data, UrbanPy can provide a detailed assessment of access to key services, such as food and health, and offers an optimization-based approach to improve this accessibility.

This paper is structured as follows. Section 2 provides a functional analysis and a comparison with similar tools. Section 3 presents a case study of the application of UrbanPy in Lima, Peru, as part of municipal efforts to address the impacts of the COVID-19 pandemic. Finally, Sect. 4 discusses the implications of UrbanPy and its future direction.

2 Main Functionality and Modules

To address the challenges that arise from working with large and diverse spatial urban data sets, UrbanPy provides a tool for the collection of data, aggregation into spatial units, and processing of road network statistics within these units consistent, automatable, and easy to use primarily for urban and transportation planning purposes.

UrbanPy was originally conceived as part of a collaboration between the Municipality of Lima, the Inter-American Development Bank, and the Universidad del Pac'ıfico, as a way to improve policy responses to the COVID-19 pandemic in Lima, Peru. The first 'test' application of UrbanPy in this city shows that, beyond urban and transport planning, the tool can also be applied for health and social protection policy decisions (see Sect. 3 below).

UrbanPy is an open-source Python package that downloads administrative boundary geometries, street networks from OpenStreetMap, and population data from Humanitarian Data Exchange. It wraps graph download and graph statistic computation via OSMnx (Boeing 2017), spatial operations with R-tree and Geopandas (GeoPandas developers 2020) and computes distance matrices and travel times using different service providers (open-source and proprietary, as preferred by users).

This library enables users to easily construct, project, visualize, and analyze several layers of data by constructing a city's or neighborhood's regular polygon segmentation, as well as downloading walking, driving, or biking networks within these polygons. As mentioned before, UrbanPy was built using Python because of its easy of use and the growing strength of its spatial data analysis libraries stack. UrbanPy is built on top of Python's NetworkX, Matplotlib, Geopandas, OSMnx and H3 libraries for rich spatial analysis capabilities, fast spatial queries and operations with R-tree indexing and crisp visualizations. This section discusses UrbanPy's primary features in order.

2.1 Downloading Spatial Data Sets

As described above, UrbanPy provides an interface to download different spatial datasets. In its current release,[1] UrbanPy provides the following methods for data download: *download.nominatim osm*, *download.hdx dataset*, *download.hdx fb population*, *download.overpass_pois*, and *download.osmnx graph*. As their names indicate, the Nominatim and Overpass APIs are supported, as well as graph download using OSMnx and population data sets using Humanitarian Data Exchange (HDX).

For population layers, using HDX, Facebook high resolution population estimates may be downloaded and turned into DataFrame structure using *download.hdx fb population*. This representation allows the user to harness both query operations (filtering, grouping, sorting) and spatial operations (intersection, spatial filtering, etc). It is important to note that HDX data can be downloaded with a specific function for Facebook data, but any resource that is stored on HDX as a compressed CSV (or just a CSV) can be downloaded using *download.hdx dataset*.

The second layer that is supported is the road network graph. UrbanPy provides three main ways to download these graph: by point (a lat, lon coordinate with a buffer radius), by place name (a string address) or based on a polygon bounding box using *download.osmnx graph*. The latter allows for road network coefficients (*e.g.* betweenness centrality, page rank) to be computed for specific geographical partitions.

The third information layer, which enables local filters using spatial operations, is *download.nominatim osm*. It provides an interface with the Nominatim API by providing a query string (*i.e.* Los Angeles, California) and an optional expected

[1] Source code may be found at https://github.com/EL-BID/urbanpy, as well as through the IDB's Code for Development platform, code.iadb.org

position (the polygon is not always located as the first result). This function queries the API and converts the polygon result into a GeoDataFrame, with EPSG 4326 as its coordinate reference system.

Finally, UrbanPy provides an interface with the Overpass API through the *download.overpass_pois* function. A set of queries for food supply, healthcare facilities, education centers, and financial establishments are supported *a priori* with the automatic construction of a DataFrame based on the result. Aside from these pre-built queries, a custom query string may be supplied, where the corresponding JSON response is returned for the user to process into the format they prefer. This function can also create GeoDataFrames for points and polygons (including linestrings) for spatial filters to be performed later on.

2.2 Constructing Hexagons While Providing Spatial Aggregation and Down-Sampling

To analyze a city in standardized spatial units, as mentioned before, several approaches can be followed. UrbanPy provides an interface with Uber's H3 hexagon partitioning system (Uber 2020). H3 was chosen to provide a regular polygon partition of a city that allows higher resolution in urban areas and lower resolution in peripheral areas, accelerating distance matrix computation and spatial aggregation procedures. Given a polygon representing a city's administrative boundaries, UrbanPy is able to partition the area to an input resolution (see H3's documentation[2] for detail into resolution, hexagon area and number of unique indexes).

Once a base hexagon layer is created, UrbanPy can be used to aggregate data from lower resolutions (points and polylines) into hexagons or perform overlay and spatial operations to downsample data from a higher resolution (administrative boundaries, census zones). These merging and aggregation operations are flexible – both the aggregation function (sum, min, max, mean, mode) and the spatial operation (intersection, union, overlay) are user-determined. This enables different metrics to be aggregated in different ways, reducing code footprint and delivering consistent results.

2.3 Constructing Distance and Travel Time Matrices

Within the routing module, UrbanPy provides support for several routing engines. If the user has docker and wget already installed, UrbanPy provides an interface to download, install and process a road network using OSM and OpenSource Routing

[2] https://h3geo.org/docs/core-library/restable

Machine (OSRM). With this processing, OSRM is capable of handling distance matrix and point-to-point path computations.

If a service provider is preferred, UrbanPy provides interfaces with the Google Maps Distance Matrix and OpenRoute Service for distance matrix computation (with a user provided API key). Note that NetworkX is also used for graph shortest path finding, whether it is used to compute the number of nodes in a path or the total length.

To compute these matrices, only the geometries are needed (in EPSG 4326). Latitude and longitude pairs are also accepted, as well as data structures other than DataFrames such as lists of (lat, lon) tuples. This added flexibility lets the user read coordinate data from different sources, merge them into a single structure and compute both distance and travel time matrices with the service of choice, letting UrbanPy handle the abstraction into API specific patterns.

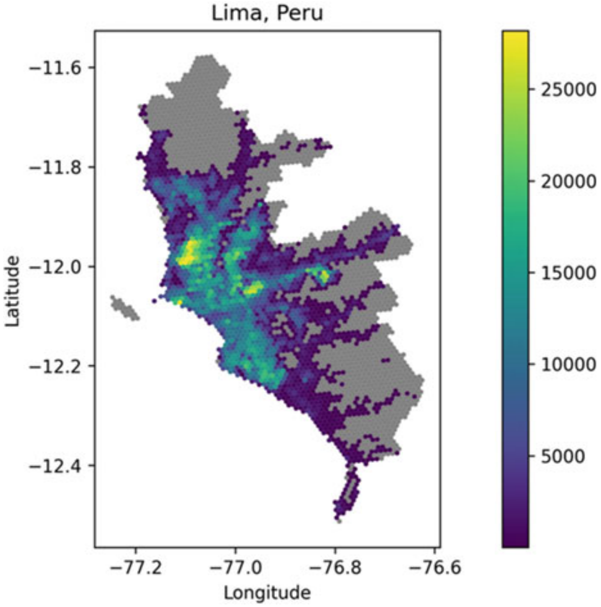
3 Case Study: Lima, Peru

Lima, Peru's capital and largest city, concentrates 30% of the country's population (11 million inhabitants, INEI (2018)). Once the COVID-19 pandemic hit the country, the national government introduced a strict quarantine and a curfew to 'flatten the epidemic curve' and mitigate the long-term impact on the economy. During this mandatory quarantine, guaranteeing a high level of accessibility to food supplies (e.g. markets, nanostores, convenience stores, and similar facilities) without risking vulnerable populations, such as the elderly, became a significant challenge. The need to optimize service delivery was a common challenge among cities in Latin America, and, as a result, the IDB and the UP launched a collaboration to design digital tools to support municipal authorities. UrbanPy was thus conceived as a tool to satisfy the need for rapid access to accurate and useful information for local authorities and support data-driven decision-making. The tool was developed through an iterative solution design process, in which the core development team and non-technical decision-makers from Lima's municipality where involved.

In this section, the process of sourcing data, converting it to spatial units and applying an optimization model to allocate itinerary markets will be presented as a use case for UrbanPy. The first stage in this initiative was to source different layers of data. To estimate demand an population flow, HDX was chosen due to the high resolution of the data collected from Facebook. These points have been filtered to Lima's administrative boundaries but due to their high resolution, they need to be aggregated for comparable analysis.

Since each record in this data set if of point geometry, a spatial unit to aggregate the data and produce relevant visualizations is needed. Thus, Lima's administrative boundaries were split into resolution 8 hexagons. Figure 1 presents the initial HDX data aggregated to hexagon level. This representation is particularly useful to aggregate other sources of data (i.e. socio-economic variables, where the median

Fig. 1 Hexagon population density layer (for 2020)



income level can be represented as the percentage of city blocks within the hexagon that perceive the median income).

Using these layers and UrbanPy's POI download functions to obtain food facilities within the city, food accessibility metrics can be measured using the estimated travel time from each spatial unit to the facilities. Then, it was determined that the current food facilities were not capable of successfully reaching the most densely population zones in the city. Also, in Lima, traditional markets are especially critical agglomeration points, which leads to a potential increase in the spread of COVID-19 among vendors and customers. In this context, parks and sport fields arise as potential relief spaces to allocate temporal food facilities. These facilities, commonly referred to as itinerary markets, may help to improve the food access and to reduce the infection rate among the daily customers in current markets and supermarkets.

A Facility Location Model was proposed to obtain the optimal itinerary markets locations. Results were delivered to different departments within the Municipality to support real-time decision making. The data acquisition and modelling process is described as follows: First, a query to the Overpass API was made to get all the candidate itinerary markets (parks, sport fields, etc). These initial results are concatenated with the national market census, while filtering to a specific district to reduce the search space for the model (as to make it computationally viable). Then, a distance matrix can be computed to serve as input to the optimization model. In Fig. 2, the filter to San Juan de Lurigancho (Lima's most densely populated district) is presented. This district will be the focus of this case study given its high levels of population and available markets in the candidate set.

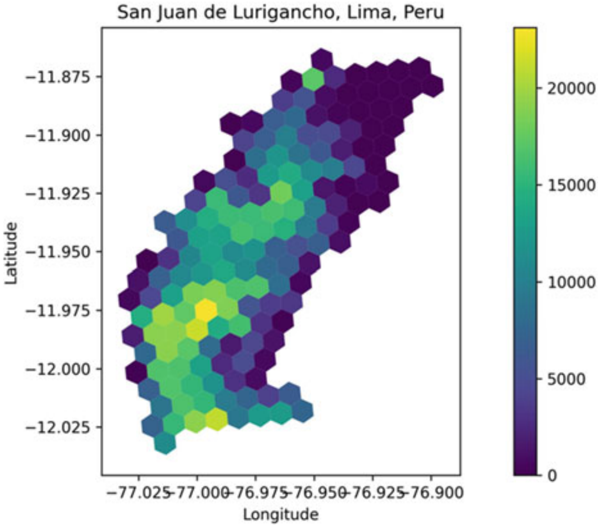


Fig. 2 Population layer for San Juan de Lurigancho, Lima, Peru

In particular, a modification of the p-median problem was applied, adding capacity constraints to the facilities and forcing the optimizer to keep certain facilities active.

The p-median problem uses two main decision variables: x_{ij} representing the percentage of the demand from customer j assigned to facility i, and y_i representing the activation of the i-th facility. As its parameters, d_j represents the demand of customer j, c_{ij} represents the cost of travelling to facility i by customer j. Finally, the number of facilities to activate must be specified using p.

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} d_j c_{ij} x_{ij} \tag{1}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \forall j \in \quad \text{Customer} \tag{2}$$

$$\sum_{i=1}^{n} y_1 = p \tag{3}$$

$$\sum_{j=1}^{m} d_j x_{ij} \le c_1 \forall i \quad \text{Facilities} \tag{4}$$

$$x_{ij} \le y_i \forall i \in \quad \text{Facilities} \quad \forall i \in \quad \text{Customers} \tag{5}$$

$$x_{ij}, y_i \in \{0, 1\} \forall i \in \quad \text{Facilities} \quad \forall i \in \quad \text{Customers} \tag{6}$$

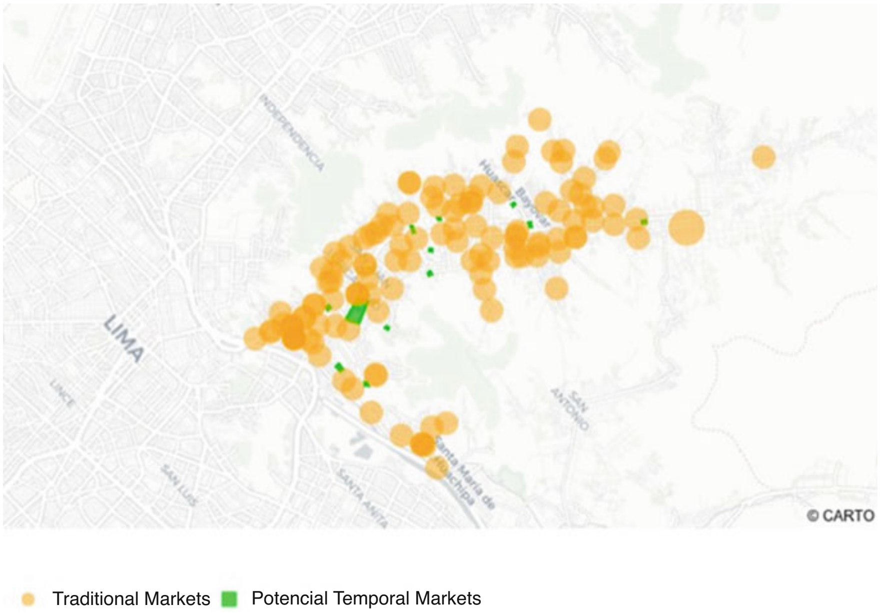


Fig. 3 Optimization results, green points and polygons represent the selected facilities and orange circles represent the active markets

$$0 \leq c_{ij} \leq 1 \forall i \in \quad \text{Facilities} \quad \forall i \in \quad \text{Customers} \tag{7}$$

$$d_j \geq 0 \forall j \in \quad \text{Customers} \tag{8}$$

Equation 1 presents the objective function: minimizing the cost of assigning $x_{ij}\%$ of the demand to the active facility. In Eq. 2 an equality constraint is enforced to ensure that for each customer, the sum percentages equals 1 (conserving the total demand for each customer). Using Eq. 3 the total number of activated facilities is constrained to p. Constraint 3 (Eq. 4) sets the capacity constraints. Since the demand is expressed in people attending a market, summing $d_j x_{ij}$ and limiting it to c_i is adequate to express this restriction. Eq. 5 links both decision variables while the last three constraints (Eqs. 6, 7 and 8) set the decision variable bounds and the cost and demand non-negativity constraints.

Finally, the results of this optimization[3] can be visualized in Fig. 3 (see the library's examples folder[4] for detail on how the data was obtained and transformed). The result of minimizing the total amount of people assigned to each candidate facility yielded some interesting insights. First, adding a constraint that ensures all

[3] A web-based version of these results is available at https://www.observatoriodelima.org/mercados_itinerantes.

[4] https://github.com/EL-BID/urbanpy/tree/master/notebooks

wholesale and retail markets remain active, ensures that all candidate activations serve the purpose of redistribution the demand of the highest population hexagons.

Another point is that a couple of football pitches and parks were selected deep inside the district, which, though located within hard access zones, still represent a valuable addition to guarantee that at-risk population does not engage in long distance travel to acquire supplies for their quarantine.

Finally, these results encourage a large scale location of itinerary markets. Even though an optimal solution may be computationally challenging (there are about 13,000 candidate facilities and 3200 hexagons), an approximate solution using lagrange relaxation or genetic algorithms can be viable for a public policy approach.

4 Discussion

In this paper, UrbanPy was presented as a library to download, process and visualize high resolution urban data. It provides support for population, administrative boundaries and POI data download, regular polygon partitioning of administrative boundaries, spatial operations for aggregation and down-sampling, routing APIs, and visualization functions.

Within the case study, UrbanPy showed its critical role in providing data to the optimization model in the context of the COVID-19 pandemic. This was done by using all the available modules to construct the candidate and customer sets for the FLP as well as the cost matrix and the demand vector. This application also showed UrbanPy's seamless integration into a Pandas/Geopandas workflow, such as incorporating data from external sources to complement the optimization's constraints.

UrbanPy can still accommodate additional features. First, UrbanPy is currently shallow in down-sampling methods (such as k-ring smoothing). When merging data from sources larger than the hexagon's resolution, the approach for merging and downsampling (how data is assigned to each hexagon) is critical for coherent and consistent analysis. Within the plotting functions, currently UrbanPy has support for Plotly choropleth maps, while the standard hexagon output is compatible with Deckgl and Pydeck. To produce some of the figures shown in this paper, Geopandas and Pandas were harnessed for their plotting features, which by design is arguably the cleanest way to produce static plots. Still, there are avenues of support for other web-based visualizations such as Bokeh, D3, among others.

Finally, is important to note that, as mentioned, UrbanPy stemmed from a collaboration between the IDB, Universidad del Pacifico and the Municipality of Lima, which allowed for both an iterative design process and a quick and rapid application of the tool. For the tool to truly enable data-driven governance, in the context of COVID-19 as well as given the twin challenges of urbanization and rapid technological change, it is essential that UrbanPy be scaled up and tested in additional cities, among a similar diverse set of stakeholders, and for new policy questions, all while maintaining a balance between a user-friendly experience, high-performing code, and rapid-deployment experience.

References

Batty, M. (2008), 'The size, scale, and shape of cities', *science* 319(5864), 769–771.

Boeing, G. (2017), 'Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks', *Computers, Environment and Urban Systems* 65, 126–139.

DOMO (2020), 'Data never sleeps 5.0', Available at https://web-assets.domo.com/blog/wp-content/uploads/2017/07/17_domo_data\-never\-sleeps\-5\-01.png. Accessed 10 July 2020.

Duranton, G. & Puga, D. (2014), The growth of cities, *in* 'Handbook of economic growth', Vol. 2, Elsevier, pp. 781–853.

GeoPandas developers (2020), 'Geopandas'. URL: https://geopandas.org

Glaeser, E. L. (1999), 'Learning in cities', *Journal of urban Economics* 46(2), 254–277. GSM (2018), 'The mobile economy 2018', *London: GSM Association*.

INEI (2018), 'Peru: Crecimiento y distribuci´ on de la poblaci´ on, 2017. primeros resul-´ tados'.

OECD. Publishing (2015), *The metropolitan century: Understanding urbanisation and its consequences*, OECD Publishing.

Townsend, A. & Zambrano-Barragan, P. (2020), 'Big urban data: A strategic guide for cities', *Washington, DC: Inter-American Development Bank*.

Uber (2020), 'H3: A hexagonal hierarchical geospatial indexing system'. URL: https://github.com/uber/H3

UN-Habitat (2016), 'World cities report', *Nairobi: UN-Habitat*.

Woetzel, J., Remes, J., Boland, B., Lv, K., Sinha, S., Strube, G., Means, J., Law, J., Cadena, A. & Von der Tann, V. (2018). 'Smart cities: Digital solutions for a more livable future', *McKinsey Global Institute: New York, NY, USA* pp. 1–152.

World Bank (2020), 'Databank – world development indicators', Available at https://databank.worldbank.org/source/world-development-indicators. Accessed 10 July 2020