Foundations for nighttime lights data analysis

Ayush Patnaik, Ajay Shah, Susan Thomas

XKDR Forum, Mumbai, Maharashtra, India Authors contributed equally to this work.

Current Address: Samarpan Complex, Andheri East, Mumbai, Maharashtra, India
* ayushpatnaik@gmail.com

Abstract

Nighttime lights captured from satellites has emerged as an important way to measure prosperity. Every researcher who uses the files released by NASA and NOAA faces the challenge of pre-processing it in addressing data quality issues. We present **NighttimeLights.jl**, a package written in Julia, which implements conventional and novel methods for cleaning the data. The package also serves as a platform for methodological research in remote sensing.

Introduction

A remarkable development in the field of alternative data is the use of satellites that measure nighttime light radiance. From the early 1990s, nighttime lights has been used to measure economic activity. Where the gross domestic product (or GDP) of a country or region is observed accurately, it correlates well with nighttime lights data [1], thus validating the new measure.

There are two situations where nighttime lights is a superior measure. First, nighttime lights has accuracy, latency and geographical resolution that is superior to conventional methods of measuring GDP (for example, [2]). Second, this alternative data is particularly useful in less developed countries, where the institutional capacity for conventional economic measurement faces limitations of state capacity. As an example, nighttime lights has been used to study the effect of land zoning, and its spillover effects, in India [3]. Similarly, nighttime lights has been used to study the effect of highways on economic activity in India [4]. Such applications are not possible with conventional data.

Dataset	Source	Period	Frequency	Resolution
DMSP-OLS	DoD-NOAA	1992 - 2013	Annual	30″
VIIRS	NASA-NOAA-DoD	April 2012 - Present	**Monthly**	**15″**

Table 1. Main data sources for nighttime lights

There are two datasets for nighttime lights which are widely used. DMSP-OLS is an annual dataset that is available from 1992 to 2013. This was discontinued after the launch of Suomi-NPP which has the "Visible Infrared Imaging Radiometer Suite" (VIIRS) sensor with superior capabilities. The two data sources are summarised in Table 1 and their differences are highlighted in [5].

	$(nW\,cm^{-2}\,sr^{-1})$	
	Radiance	Cloud-free observations
Minimum	-0.36	0
Mean	0.61	8.33
Median	0.17	9
99^{th} percentile	7.80	17
Maximum	65.10	18
Number of measurements	$249 \times 157 \times 25$	$249 \times 157 \times 25$
$(height \times width \times number\ of\ months)$		

Table 2. Summary statistics about the raw data for the box of VIIRS nighttime lights around Goa

NOAA computes the monthly average radiance of VIIRS nighttime lights for each pixel, within the set of days that are considered cloud-free. Their data release consists of the mean radiance, and the number of days which were considered cloud-free and thus went into the computation. This dataset, which we term 'raw nighttime lights data', is the starting point for all researchers. It has numerous and well-known problems such as background noise, outliers, and negative values. Every researcher in this field has to deal with these issues using a variety of algorithms. This essential pre-processing stage constitutes a problem in the research process; it induces an entry barrier that hinders the utilisation of nighttime lights, and induces non-comparability and non-reproducibility across research projects.

In this paper, we introduce an open-source software package, **NighttimeLights.jl**, which performs cleaning procedures on the VIIRS monthly nighttime lights using statistical methods from the present research frontier. Through this, we solve the problems of entry barriers, non-comparability and non-reproducibility.

1 Problems of using raw nighttime lights data

There are five known problems with the nighttime lights data. These are (i) missing data, (ii) negative radiance values, (iii) outliers, (iv) background noise and (v) attenuation of radiance in cloudy months.

In this section, we show a sample dataset that will be used throughout this paper, and demonstrate each of these problems in this sample dataset. The sample dataset is the monthly VIIRS nighttime lights from April 2012 to April 2014, of a box around the Indian state of Goa, which has 39,093 pixels of radiance values. We refer to this as a sample of 'raw data'. The steps for downloading the data and cropping it are at S1 Appendix. Summary statistics about this sample are in Table 2. We emphasise that the raw data, and the summary statistics thereof, pertain to the data directly from NASA/NOAA.

Problem 1: Missing data. The raw data contains the mean radiance computed over cloud-free days. The mean is reported as missing when all days of the month are cloudy, and is coded as '0' by NOAA. As an example, in July 2014, 69.75% pixels in our sample dataset had this missing value, i.e. in these pixels there were 0 days in this month that were deemed to be cloud-free by NOAA.

Problem 2: Negative values. While radiance should only be positive, negative values are also present in the raw data. These are due to erroneous calibration of radiance in the presence of air glow [6]. This is not just an occasional problem; in December 2012, a full 57.48% of the pixels in the Goa dataset were negative.

Problem 3: Outliers. A few very large values for radiance in the raw data are likely to be due to gas flares, fires, etc [7]. These outliers constitute noise in the measurement of economic activity. Across all pixel-months of radiance observations for the Goa dataset, the mean value is 0.61 $nW\,cm^{-2}\,sr^{-1}$, and the 99^{th} percentile value is 7.88 $nW\,cm^{-2}\,sr^{-1}$, but the maximum value is 65.10. These outliers can be classified into two categories:

1. There are pixels with occasional extreme values. These could potentially be associated with measurement errors or physical phenomena like fires.
2. There are pixels with very high mean and very high variance. Such pixels tend to have an industrial explanation, such as flaring of gas.

Problem 4: Background noise. Places with no economic activity may show some low, but non-zero values due to background noise [8]. These small values induce errors in computing zonal statistics as small values add up. For example, consider a forest or a desert, with no economic activity, that has the same area as the Goa dataset. If the background noise observed in each pixel is a plausible value of $0.4nW\,cm^{-2}\,sr^{-1}$, the aggregate radiance over this area works out to $15{,}637.20nW\,cm^{-2}\,sr^{-1}$. This value is significant when compared with the observed aggregate radiance in our sample data, in April 2012, of $27{,}410.29nW\,cm^{-2}\,sr^{-1}$.

Problem 5: Bias in cloudy months. There is a downwards bias in the reported radiance when the number of cloud-free images in a month is low [9]. This bias is known to take on large values ranging from -10% to -30% in cloudy months.

A fuller exposition of these five problems, and alternative solutions to deal with them, is in [9]. Our focus here is on the open-source package which implements these methods.

NighttimeLights.jl

NighttimeLights.jl is written in **Julia**, in order to obtain the best performance. The package is built on top of **Rasters.jl**, a package to read and process geospatial data. **NighttimeLights.jl** can found at `https://github.com/xKDR/NighttimeLights.jl`.

The source data format

Nighttime lights images are provided in the form of TIF files. These are read as 2D arrays using the **Rasters.jl** package. Images taken at different times are stacked together to form 3D arrays using the `Rasters.combine` function.

In the examples ahead, we use two data cubes: (1) `goa_radiance` is the data cube of the monthly mean radiance values for each pixel, and (2) `goa_n_cloudfree` is the data cube of the number of cloud-free observations which were used in computing the mean radiance of each observation.

Function naming scheme

The different components of each function name are separated by underscores. The name of a function begins with the problem that it intends to address. For example, all functions that deal with problems associated with NAs have a name that starts with `na`; the functions that perform interpolation for NAs have a name that starts with `na_interp`; and the function which does linear interpolation for NAs is called `na_interp_linear`.

Cleaning methods

1. Dealing with missing data

   There are two functions to deal with missing data: `na_recode` and `na_interp_linear`.

   (a) `na_recode` examines the number of cloud-free observations that go into computing the mean radiance for a pixel, and when no observations are found, the 0 value for radiance (that comes from NOAA) is recoded to `missing`. It takes a radiance data cube and its corresponding cloud-free observations data cube as input. It returns a radiance data cube with the missing values correctly marked as `missing`.
   Example: `na_recode(goa_radiance, goa_n_cloudfree)`.

   (b) `na_interp_linear` uses linear interpolation over the time-series of a pixel to replace `missing`s.
   The function accepts a vector of radiance for one pixel as input. If the first or the last element of the time series is `missing`, it chooses the value of the first non-missing neighbor. It returns the input vector where all `missing` values have been replaced by real values.
   Example: `na_interp_linear(goa_radiance[1,1,:])`.
   It can be applied to all pixels of a data cube using `long_apply`.
   Example: `long_apply(na_interp_linear, goa_radiance)`

2. Dealing with negative values

   There is one function to deal with negative values of the radiance: `neg_replace`.

   It accepts a data cube as input, replaces all negative values with a given replacement value, which is `missing` by default, and returns the modified data cube.
   Example: `neg_replace(goa_radiance; replacement = missing)`

3. Dealing with outliers

   There are two functions to deal with outliers: `outlier_hampel` and `outlier_variance`.

   (a) `outlier_hampel` marks the extreme observations within the radiance time-series for a pixel as `missing`.
   The function accepts a vector of the time-series of the radiance for one pixel as input. It identifies outliers using a Hampel filter [10] with default settings (window width of 5, and outlier classification outside 3 standard deviations). These outliers are recoded as `missing` in the returned object.
   Example: `outlier_hampel(goa_radiance[1,1,:])`
   It can be applied to all pixels of a data cube using `long_apply`
   Example: `long_apply(outlier_hampel, goa_radiance)`

   (b) `outlier_variance` creates a mask of pixels with high variance.
   The function accepts a data cube as input. For each pixel, it computes the variance of the time-series of detrended radiance for each pixel. Pixels with variance above a provided threshold are considered outliers. This threshold is provided as a number between 0 and 1, representing a quantile. The threshold defaults to 0.999th quantile. The function returns a 2D binary matrix where the values are 0 if the pixel has outliers and 1 otherwise.
   Example:

```
outlier_variance(goa_radiance; threshold = 0.999)
```

The pixels with high variance can be zeroed out using `apply_mask`.
Example:

```
apply_mask(outlier_variance(goa_radiance,
          goa_n_cloudfree), goa_radiance)
```

4. Dealing with background noise

   The package has one function to filter background noise: `bgnoise_PSTT2021`.

   This implements the background noise removal algorithm described in [9]. It accepts a radiance data cube and its corresponding data cube for the number of cloud-free observations as input. It computes an annual image using the last year of the data by computing the weighted average of radiance of each month of the year by using cloud-free observations as weights [11]. The pixels in this annual image with values below a threshold are considered dark. The default value for the threshold is 0.4 $nW\,cm^{-2}\,sr^{-1}$.
   Example:
   `bgnoise_PSTT2021(goa_radiance, goa_n_cloudfree; threshold = 0.4)`
   The function returns a 2D binary matrix, where zeroes represent the pixels that should be considered as dark. The pixels considered background noise can be zeroed out using `apply_mask`.
   Example:

```
apply_mask(bgnoise_PSTT2021(goa_radiance,
            goa_n_cloudfree),goa_radiance)
```

5. Dealing with bias in cloudy months

   There is one function to correct the attenuation in radiance in cloudy months: `bias_PSTT2021`.

   It implements the bias correction methodology described in [9], using information about the number of cloud-free observations to partially bias-correct the radiance data. The function accepts a data cube of radiance and its corresponding data cube for the number of cloud-free observations as input. For every pixel, we estimate a non-parametric relationship (a smoothing spline) between the number of cloud-free observations and the de-trended radiance. This estimated relationship is used for bias correction of observations with small number of cloud-free observations. The function returns a data cube representing radiance.
   Example: `bias_PSTT2021(goa_radiance, goa_n_cloudfree)`

We recommend the following sequence of steps to go from raw radiance to cleaned:

1. Code missing data correctly using cloud-free observations.
2. Code negative values as missing.
3. Apply background noise removal.
4. Apply outlier rejection based on the variance test.
5. Apply outlier rejection based on time-series outlier detection.

6. Do bias correction for all pixels.

7. Use linear interpolation to fill in the missing data.

These steps, except for the bias correction stage, are widely used and are termed 'conventional cleaning' in [9]. For convenience, all these steps of conventional cleaning are brought together as a single function `PSTT2021_conventional()`.

Example: `PSTT2021_conventional(goa_radiance, goa_n_cloudfree)`

The complete set of steps – which is our recommended procedure – is implemented as a single function called `PSTT2021()`.

Example: `PSTT2021(goa_radiance, goa_n_cloudfree)`

In the future, we expect there will be further methodological advances in this field. The function `clean_complete()` will represent our views on an optimal set of steps for pre-processing in the future (for the period for which this package is actively maintained). At present, it is identical to `PSTT2021()` but in the future these could diverge.

Example: `clean_complete(goa_radiance, goa_n_cloudfree)`.

Masked nighttime lights data from NOAA in which background noise has been removed has been made available by the Earth Observation Group. Performing background noise removal on this data may not be required. In order to address this case, the keyword argument `bgnoise_clean` can be set to false in the `clean_complete` function to skip the background noise removal step.

Example of usage:
`clean_complete(goa_radiance, goa_n_cloudfree; bgnoise_clean = false)`.

A complete example

The following Julia code illustrates how one can use **NighttimeLights.jl** to go from raw data to cleaned for the sample raw data we selected as a box of nighttime lights around Goa. The simplest path is to just say:

```
goa_cleaned = clean_complete(goa_radiance, goa_n_cloudfree)
```

which is presently identical to:

```
goa_cleaned = PSTT2021(goa_radiance, goa_n_cloudfree)
```

and corresponds to the underlying steps:

```
tmp = na_recode(goa_radiance, goa_n_cloudfree)
tmp = neg_replace(tmp)
lit_pixels = bgnoise_PSTT2021(tmp, goa_n_cloudfree)
tmp = apply_mask(tmp, noise)
stable_pixels = outlier_variance(tmp, noise)
tmp = apply_mask(tmp, stable_pixels)
tmp = long_apply(outlier_hampel, tmp)
tmp = bias_PSTT2021(tmp, goa_n_cloudfree)
goa_cleaned = long_apply(na_interp_linear, tmp)
```

These three alternative pathways all apply the identical methods to transform the raw data (`goa_radiance, goa_n_cloudfree`) into a cleaned data cube `goa_cleaned`.

	Raw	Cleaned
Minimum	-0.36	0.0
Mean	0.61	0.66
Median	0.17	0.0
99^{th} percentile	7.80	9.8
Maximum	65.10	40.66

Table 3. Effect of cleaning on nighttime lights

Results

Table 3 shows some summary statistics of raw nighttime lights and cleaned nighttime lights for the Goa example. There are no negative values in the data cube after cleaning. The maximum value after cleaning is lower, as some outlier values have been removed. Half the values in the raw data cube are below 0.17, these can be attributed to background noise. All these measurements have been pushed to 0. The increase in the 99^{th} percentile value is due to the bias correction procedure.

Discussions

The VIIRS Nighttime lights is an important dataset, but every researcher who needs to use it requires an array of standard steps to pre-process the raw data from NASA/NOAA. We have built the first open-source package which implements these steps.

Methodological research on nighttime lights data continues. This package can serve as the foundation for such work. The techniques developed in [9] constitute a conservative algorithm that will always improve data quality at the price of leaving a significant amount of cloud-related bias in the data. Future enhancement of these methods will take place. Similarly, the package presently has simple techniques for outlier detection and for interpolation and does not address a pixel that is cloudy on all days of a month.

There are fundamental physical facts about nighttime lights which also require improved methods. The transition from older lamps to LED bulbs has reduced the radiance detected due to the low sensitivity of VIIRS to light with a wavelength from 400 to 500 nm. As an example, [12] shows images of Milan taken from the International Space Station. They show that when the city transitioned from conventional lighting to LED, the nighttime lights radiance declined. Given the global movement in favour of LED lighting, this introduces a problem with the use of nighttime lights data. [13] has proposed using a radiative transfer model to correct for this bias. Such new methods could be implemented in the future in **NighttimeLights.jl**.

Supporting information

S1 Appendix. In this paper, the nighttime lights data of the Western Indian state of Goa is used. The following steps demonstrate how to download raw data from Payne Institute's Earth Observation Group and crop it to a box around Goa.

1. Go to the VIIRS Monthly Nighttime Lights download page at Earth Observation Group
2. Click on `2012`, the on `201204`. This means April 2012. Then click on `vcmcfg`

3. Six files are shown. These 6 files represent the 6 tiles. Click on the file that contains the string `75N060E`. This means TILE3, which contains Goa.

4. Extract the files and put the file with `avg_rade9h` in a folder called `folder_1` and put the file with `cf_cvg` in a different folder called `folder_2`. You can use any folder names.

5. Repeat the process for all months till April 2014 The first four files of `folder_1` are listed below.

```
SVDNB_npp_20120401-20120430_75N060E_vcmcfg_v10_c201605121456.avg_rade9h.tif
SVDNB_npp_20120501-20120531_75N060E_vcmcfg_v10_c201605121458.avg_rade9h.tif
SVDNB_npp_20120601-20120630_75N060E_vcmcfg_v10_c201605121459.avg_rade9h.tif
SVDNB_npp_20120701-20120731_75N060E_vcmcfg_v10_c201605121509.avg_rade9h.tif
```

6. The following command crops and loads the radiance data cube for Goa:

```
using Rasters
using NightimeLights
bounds = X(Rasters.Between(73.67, 74.33)), Y(Rasters.Between(14.75, 15.79))
dates = collect(range(start = Date(2012,4), step = Month(1), length = 25))
timestamps = NighttimeLights.yearmon.(dates)
radiance_path = "folder_1"
filelist = readdir(radiance_path)
radiances = [Raster(i, lazy = true)[bounds...] for i in radiance_path.*filelist]
series = RasterSeries(radiances, Ti(timestamps))
goa_radiance = Rasters.combine(series, Ti)
```

   This yields a data cube of size (249, 157, 25)

7. The following command crops and loads the cloud-free observations data cube for Goa:

```
n_cloudfree_path = "folder2"
filelist = readdir(n_cloudfree_path)
n_cloudfree = [Raster(i, lazy = true)[bounds...] for i in cfobs_path.*filelist]
series = RasterSeries(n_cloudfree, Ti(timestamps))
goa_n_cloudfree = Rasters.combine(series, Ti)
```

   This yields a data cube of size (249, 157, 25)

S1 Computational details The results in this paper were obtained using **Julia** 1.7.2 using **NighttimeLights.jl** 0.6.0. **Julia** is available at `https://julialang.org/`, and **NightimeLights.jl** is available at `https://github.com/xKDR/NighttimeLights.jl`.

The package is registered in the general registry and can be installed via `using Pkg; Pkg.add( NighttimeLights)`

Acknowledgments

We acknowledge the contributions of Anshul Tayal, Siddhant Chauddhary, and Hrishikesh Saikia to the package. They have contributed to the code, and also helped us think about the design.

References

1. Doll CN, Muller JP, Morley JG. Mapping regional economic activity from night-time light satellite. imagery. Ecological Economics. 2006;57(1):75–92.

2. Gibson J, Olivia S, Boe-Gibson G. Night Lights in Economics: Sources and Uses. Journal of Economic Surveys. 2020;34(5):955–980.

3. Blakeslee D, Chaurey R, Fishman R, Malik S. Land Rezoning and Structural Transformation in Rural India: Evidence from the Industrial Areas Program. The World Bank Economic Review. 2022;36(2):488–513.

4. Ghani E, Goswami AG, Kerr WR. Highways and spatial location within cities: Evidence from India. The World Bank Economic Review. 2017;30(Supplement_1):S97–S108.

5. Elvidge CD, Baugh KE, Zhizhin M, Hsu FC. Why VIIRS data are superior to DMSP for mapping nighttime lights. Proceedings of the Asia-Pacific Advanced Network. 2013;35(0):62.

6. Uprety S, Cao C, Gu Y, Shao X, Blonski S, Zhang B. Calibration improvements in S-NPP VIIRS DNB sensor data record using version 2 reprocessing. IEEE Transactions on Geoscience and Remote Sensing. 2019;57(12):9602–9611.

7. Jiang W, He G, Long T, Liu H. Ongoing Conflict Makes Yemen Dark: From the Perspective of Nighttime Light. Remote Sensing. 2017;9(8). doi:10.3390/rs9080798.

8. Ma T, Zhou C, Pei T, Haynie S, Fan J. Responses of Suomi-NPP VIIRS-derived nighttime lights to socioeconomic activity in China's cities. Remote Sensing Letters. 2014;5(2):165–174.

9. Patnaik A, Shah A, Tayal A, Thomas S. But clouds got in my way: Bias and bias correction of VIIRS nighttime lights data in the presence of clouds. Available at SSRN 3957319. 2021;.

10. Pearson RK. Outliers in process modeling and identification. IEEE Transactions on control systems technology. 2002;10(1):55–63.

11. K Gupta P, K Srivastav S, V R Sesha Sai M, Gharai B, Senthil Kumar A, V N Krishna Murthy Y. Analysis of SNPP-VIIRS-DNB derived nightlights over India. Asian Association of Remote Sensing. 2017;.

12. Kyba CCM, Kuester T, de Miguel AS, Baugh K, Jechow A, Hölker F, et al. Artificially lit surface of Earth at night increasing in radiance and extent. Science Advances. 2017;3(11):e1701528. doi:10.1126/sciadv.1701528.

13. Ivan K, Holobâcă IH, Benedek J, Török I. VIIRS Nighttime Light Data for Income Estimation at Local Level. Remote Sensing. 2020;12(18). doi:10.3390/rs12182950.