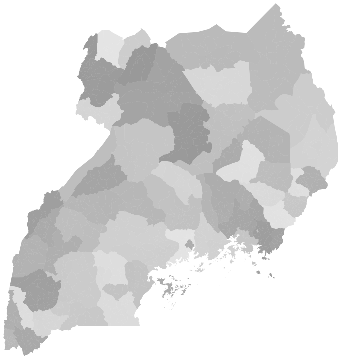
Ugandan Weather: First Pass

Joe Brew

January 24, 2015



Lendable

The following is a brief overview of a first attempt to systematically collect Ugandan weather-related data. I outline the code I used to capture the data, describe the data's quality, summarize the method's flaws, and propose some options for improving this process.

Contents

Webscraping Ugandan weather data **2**

Method 2

Results 2

Advantages and shortcomings 2

Moving forward 2

Visuals **3**

Details **5**

Webscraping Ugandan weather data

Method

I wrote some code in R to scrape historical Ugandan weather data directly from the Weather Undeground website. The scrape_data() function takes a location, start date, end date and airport boolean[1] and returns a format-standardized dataframe (R object) which can easily be written to a transportable file (.csv, .txt, etc.):

```
scrape_data(station = "ICENTRAL28",
            start_date = Sys.Date()
                          - 365,
            end_date = Sys.Date(),
            airport = TRUE))
```

The data returned consist of daily observations of precipitation, temperature, humidity, cloud cover, and dew point.

I also cobbled together a few plotting functions to visualize variation in these indicators over time and space (see charts after report).

Results

Results are mediocre. Of the 15 total Ugandan locations available in Wunderground's search options, only 4 appear to have functioning weather stations, of which 3 are in the same area (Kampala).

Data from Kampala appear to be of relatively high (usable) quality, especially if we supplement Kampala-area data to supplement on days in which data are missing. Data from Soroti (a medium-sized town about 200km to the Northwest of Kampala) are also decent. In both locations, precipitation data appear unreliable (far too low).

Advantages and shortcomings

Advantages: Multiple weather-related indicators from two areas of Uganda are readily-available and easily accessed. This method of directly scraping is simple enough that it could be implemented easily and scheduled to update regularly. This did not require use of Wunderground's API (which is relatively restrictive).

Shortcomings: This method assumes the user knows the name and type of weather stations of interest. Given that it's not API-based, it is subject to formatting changes at any time. Only two geographic locations in Uganda have valuable data. Data quality/precision has not been assessed.

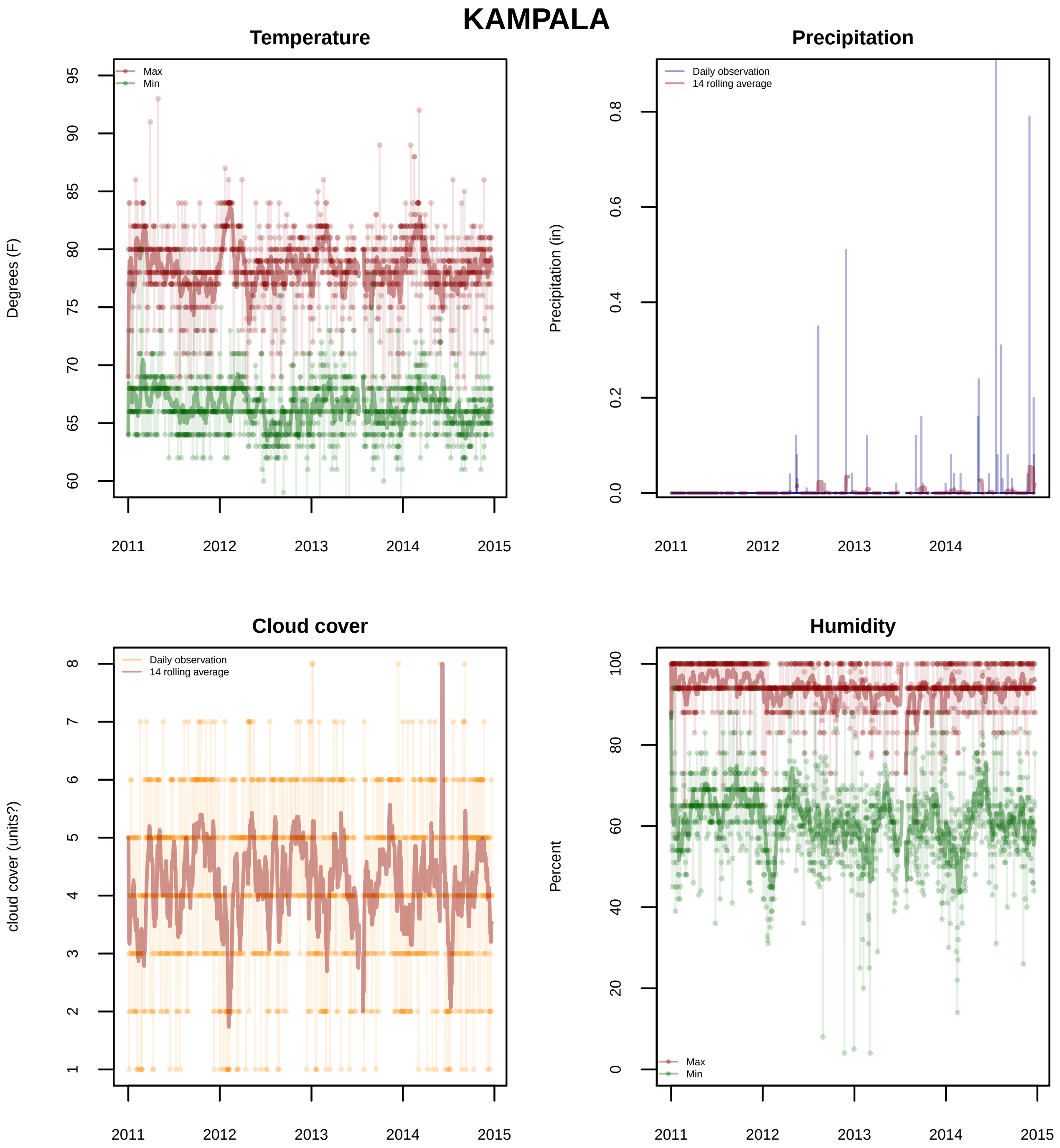
Moving forward

This first pass demonstrated that historical location-specific weather data are relatively easy to capture. Moving forward, we'll need to

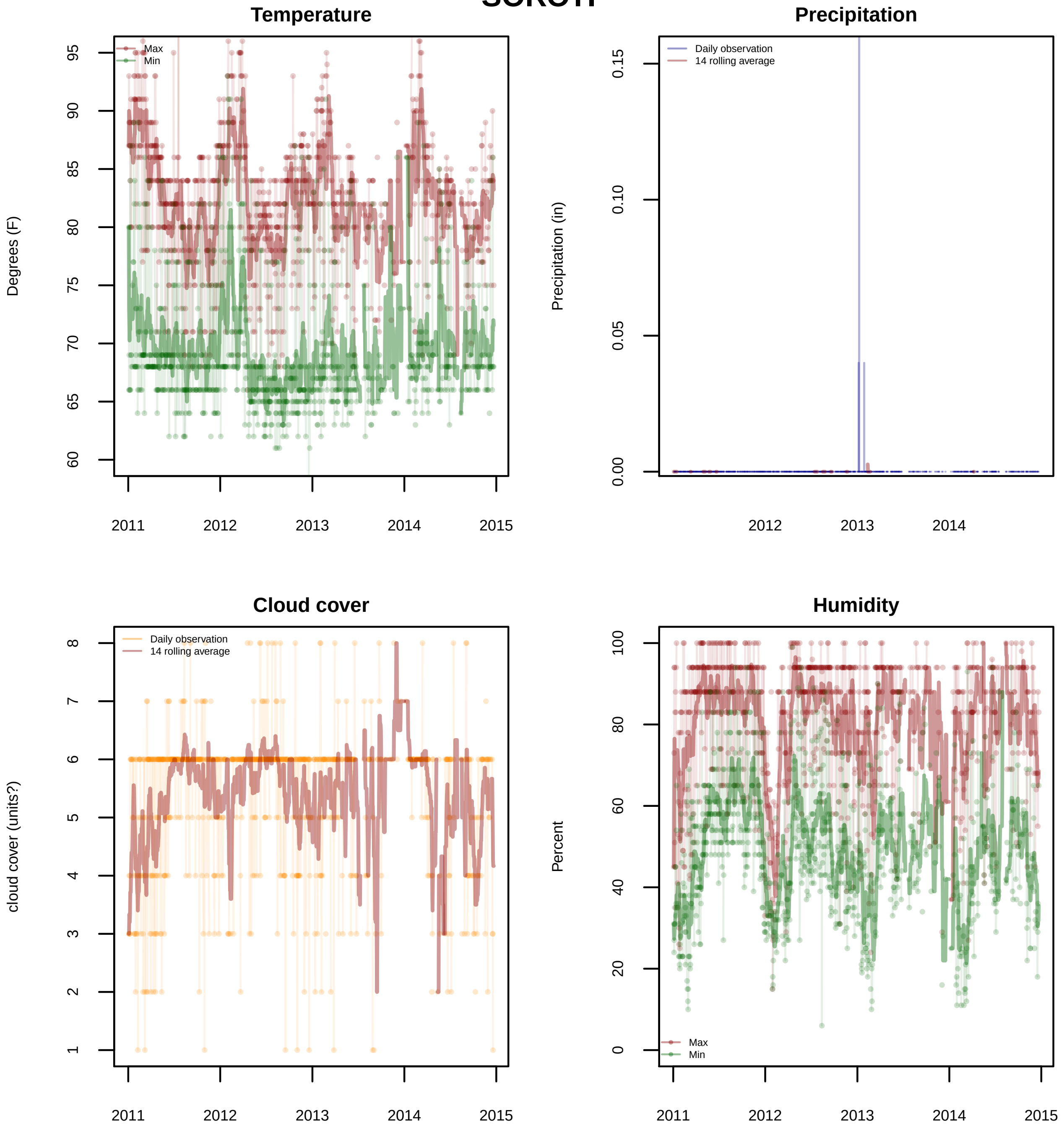
- Explore other weather data sources for more granularity within Uganda (since Wunderground is limited in this).
- Develop future code in a way that doesn't require *a priori* knowledge regarding weather station nomenclature, etc. (ie, functions to take a location and return nearest weather, with some quantification of uncertainty).
- Consider using an API and integrate weather data grabs directly into the app (necessary if we want to make weather a feature of many projects, unnecessary if we choose to incorporate weather on a project-specific basis).
- Test whether weather data is even predictive! And if it is, use caution in how we model with it - in the case of solar units, raininess might be correlated with increased risk of non-payment (less revenue generation), but severe spikes in raininess might correlate with power outages (which could be good for business).

[1] Weather Underground's URLs are oddly different for airport and non-airport locations

Visuals


KAMPALA
Temperature
Degrees (F)
Max
Min
60
65
70
75
80
85
90
95
2011
2012
2013
2014
2015
Precipitation
Precipitation (in)
Daily observation
14 rolling average
0.0
0.2
0.4
0.6
0.8
2011
2012
2013
2014
Cloud cover
cloud cover (units?)
Daily observation
14 rolling average
1
2
3
4
5
6
7
8
2011
2012
2013
2014
2015
Humidity
Percent
Max
Min
0
20
40
60
80
100
2011
2012
2013
2014
2015

SOROTI


Temperature
Degrees (F)
60
65
70
75
80
85
90
95
Max
Min
2011
2012
2013
2014
2015
Precipitation
Precipitation (in)
0.00
0.05
0.10
0.15
Daily observation
14 rolling average
2012
2013
2014
Cloud cover
cloud cover (units?)
1
2
3
4
5
6
7
8
Daily observation
14 rolling average
2011
2012
2013
2014
2015
Humidity
Percent
0
20
40
60
80
100
Max
Min
2011
2012
2013
2014
2015

Details

Full code at https://github.com/joebrew/weather/tree/master/uganda:

- Function to scrape data
- 4 years of Ugandan weather data
- Some clunky weather plotting functions
- Slightly less clunky, interactive weather-plotting functions
- Code to grab shapefiles from anywhere in the world
- Code used for this report

Weather undeground resources:

- Weather underground
- Weather underground's API page

Resources I haven't looked into in detail yet (but will):

- Openweathermap: Free historical weather API
- Worldweatheronline: looks sketchier
- Yahoo's weather API
- Page with lots of weather API resources I haven't looked into yet

This report was generated on January 24, 2015. The author used R version 3.1.2 (2014-10-31) (Pumpkin Helmet) on a linux-gnu OS.

The analysis in this report was written in the R programming language, and the report production was programmed in LaTeX using Sweave.