

SateLIFE

CS123 - Final Presentation

Beth Bailey, Joseph Denby, Cooper
Nederhood & Laurence Warner

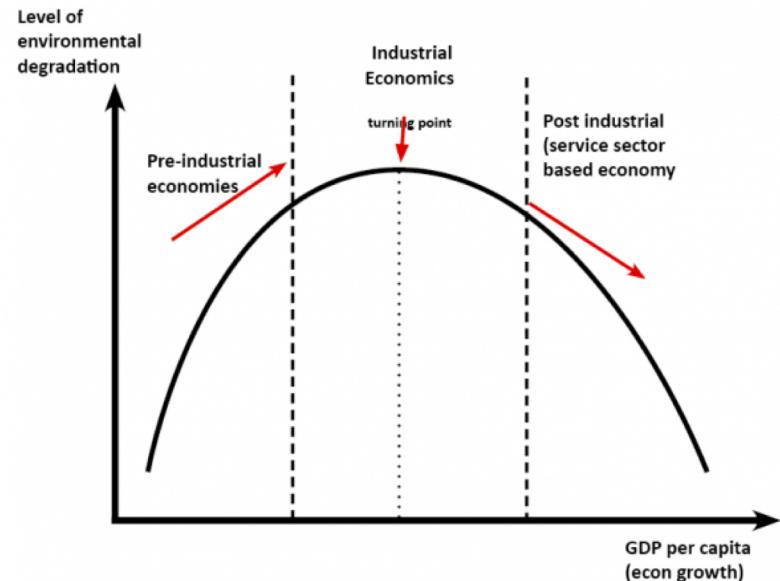
Data

- Google Earth satellite data: Brazzaville and Kinshasha
- 1.6 billion total pixels
 - 11 years of 6 bands
 - Land surface temperature
 - Nightlights
 - NDVI (vegetation index)
 - Landsat RGB (3 bands)
 - 4,900 by 5,036 pixels each
- ~2GB, but operations can be scaled up to larger sizes
- GPS coords from OpenStreetMaps



Hypothesis

- Dramatic growth and urbanization == environmental degradation
- Kuznet's Curve
 - Expect these cities to be on the left side of the curve
- Neighborhood Analysis
 - There may be different trends in different neighborhoods



Bash Scripting

Automating VM initialization

```
1#!/bin/bash
2if [[ $# -ne 1 || $1 -le 0 ]]; then
3    echo "Incorrect input: Supply one argument specifying the number of VM instances to initialize"
4else
5    location=""
6    while [[ $location != "root" ]] && [[ $location != "all" ]]
7    do
8        read -p "Specify data location ('all' or 'root') followed by [ENTER]: " location
9        if [[[ $location != "root" ]] && [[ $location != "all" ]]]; then
10            echo "Enter a valid location"
11        fi
12    done
13    for i in `seq 1 $1`; do
14        echo "INITIALIZING earth-$i..."
15        gcloud compute instances create earth-$i
16        gcloud compute scp ./chain.sh earth-$i:~/ --ssh-key-file=~/ssh/google-cloud-cs123
17    done
18    gcloud compute instances list --format=text \
19    | grep '^networkInterfaces\[0-9\]+\.\networkIP:' | sed 's/^.* //g' > hosts
20    for i in `seq 1 $1`; do
21        gcloud compute scp ~/ssh/google-cloud-cs123 earth-$i:~/id_rsa --ssh-key-file=~/ssh/google-cloud-cs123
22        gcloud compute scp hosts earth-$i:~/ --ssh-key-file=~/ssh/google-cloud-cs123
23        gcloud compute ssh earth-$i --ssh-key-file=~/ssh/google-cloud-cs123 --command="echo 'Installing dependencies on earth-$i.
24        sudo apt-get -y install git-core;
25        sudo apt-get -y install mpich;
26        sudo apt-get -y install python-pip;
27        sudo apt-get -y install python-dev;
28        sudo apt-get -y install python3-dev;
29        sudo apt-get -y install python3-pip;
30        yes | sudo pip install mpi4py;
31        yes | sudo pip3 install mpi4py;
32        yes | sudo pip install numpy;
33        yes | sudo pip3 install numpy;
34        yes | sudo pip3 install scikit-image;      1  #!/bin/bash
35        yes | sudo pip install scikit-image;      2  for i in $(cat hosts); do
36        yes | sudo pip install pandas;           3  ssh -oStrictHostKeyChecking=no $i "echo connected to $i"
37        yes | sudo pip3 install pandas;          4  done
38        echo 'Initiating chaining from earth-$i';  bash ~/chain.sh";
39    done
40    if [[ $location == "root" ]]; then
41        echo "Sending to root..."
42        gcloud compute scp --recurse data earth-1:~/ --ssh-key-file=~/ssh/google-cloud-cs123
43    else
44        echo "Sending to..."
45        for i in `seq 1 $1`; do
46            echo "...node $i"
47            gcloud compute scp --recurse data earth-$i:~/ --ssh-key-file=~/ssh/google-cloud-cs123
48        done
49    fi
50    gcloud compute ssh earth-1 --ssh-key-file=~/ssh/google-cloud-cs123
51
52fi
```

Data Analysis

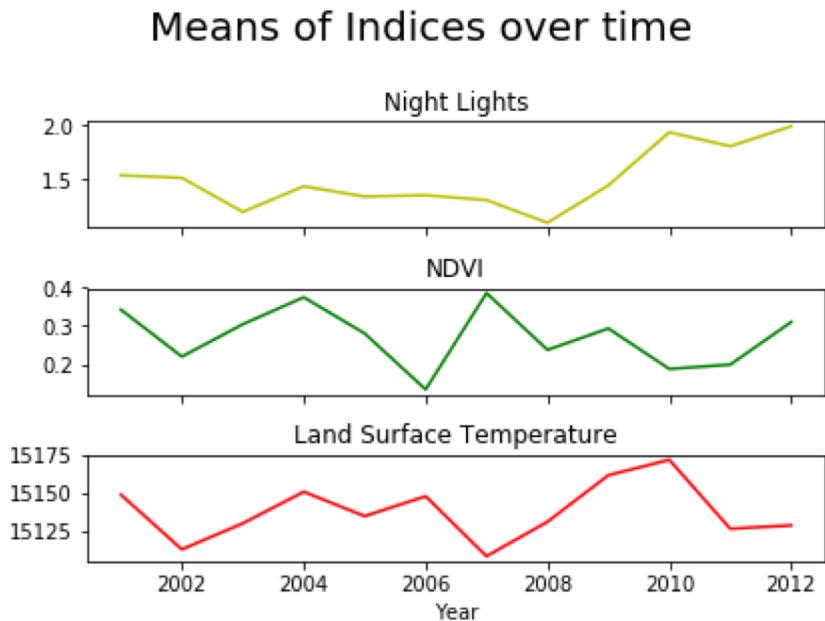
MPI and Numpy

MPI

- Data in tiff file format, used class to build into 3D Numpy arrays
 - Size = [4,900, 5,036, # of bands]
 - Also array of neighborhood boundaries (size = [4,900, 5,036])
 - Scattered chunks of these arrays to nodes
- Used MPI (scatter, gather, broadcast) to do operations on each node
- Numpy operations are very efficient (sum, max, min, etc.)
 - Some analysis of all the pixels took less than 5 minutes (faster using MPI)
 - Neighborhood analysis harder
 - Looped over data
 - Used dictionaries to store neighborhood values
 - ~1.5hrs on local machine with two cores for our dataset

Results & Visualization

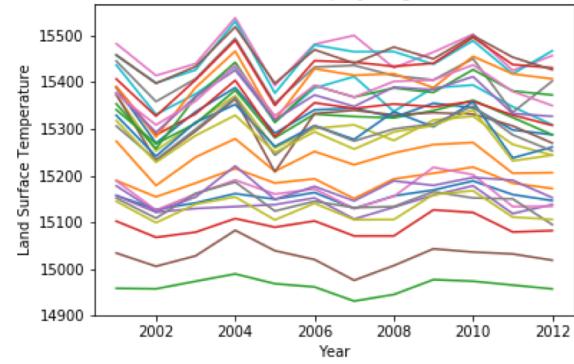
Section 1. Area-wide analysis of different bands



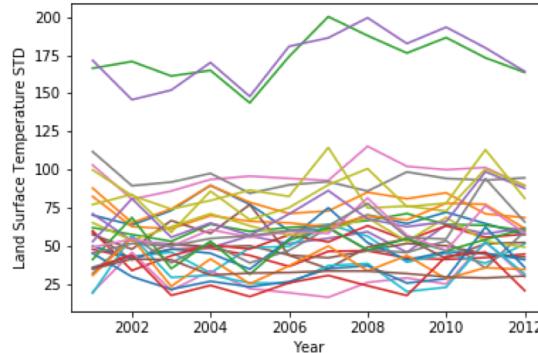
Section 2. Neighborhood Comparison

LAND SURFACE TEMP

Land Surface Temp by Neighborhood

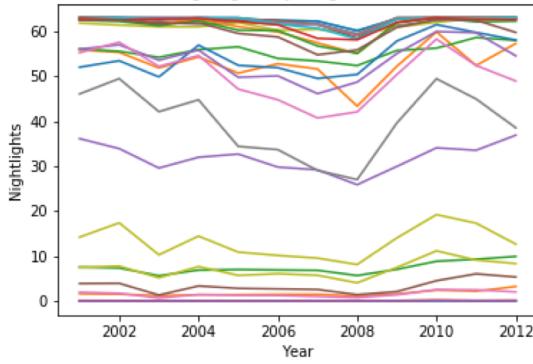


Land Surface Temp STD by Neighborhood

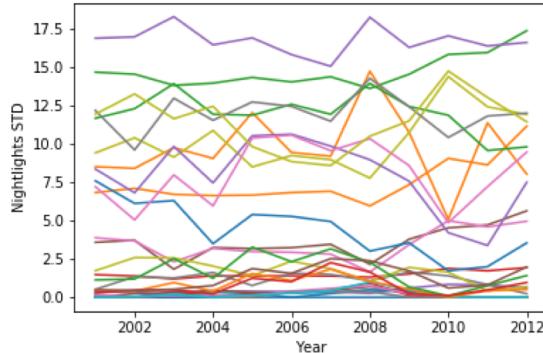


NIGHTLIGHTS

Nightlights by Neighborhood

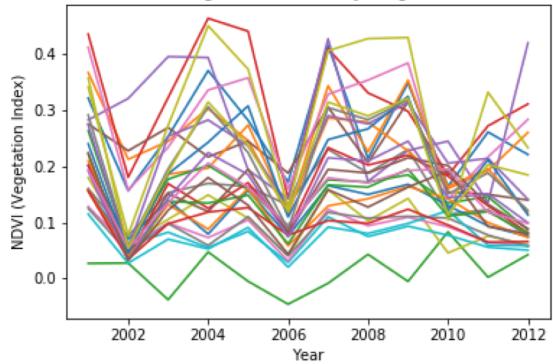


Nightlights STD by Neighborhood

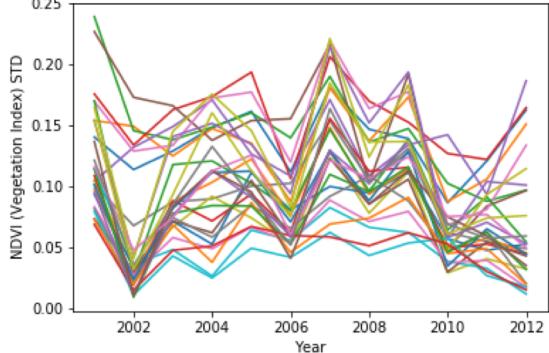


NDVI

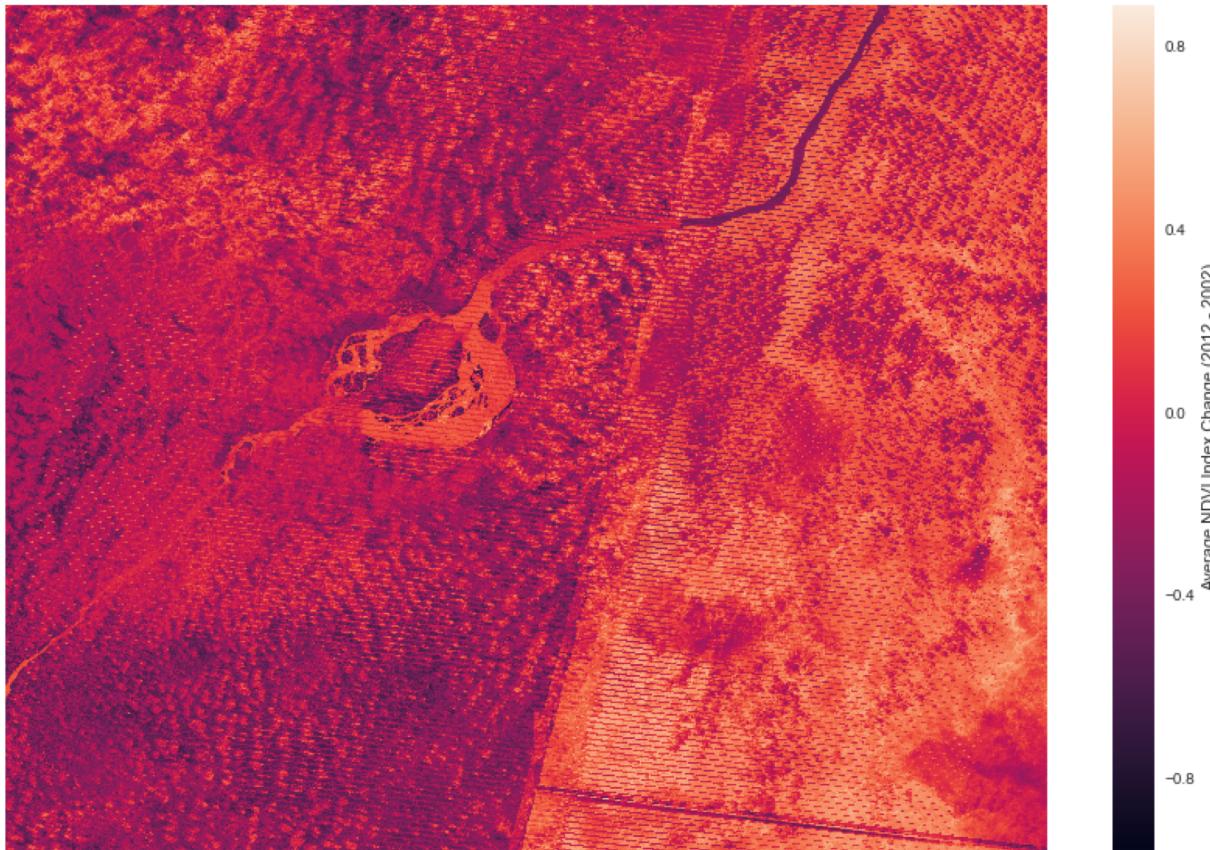
NDVI (Vegetation Index) by Neighborhood



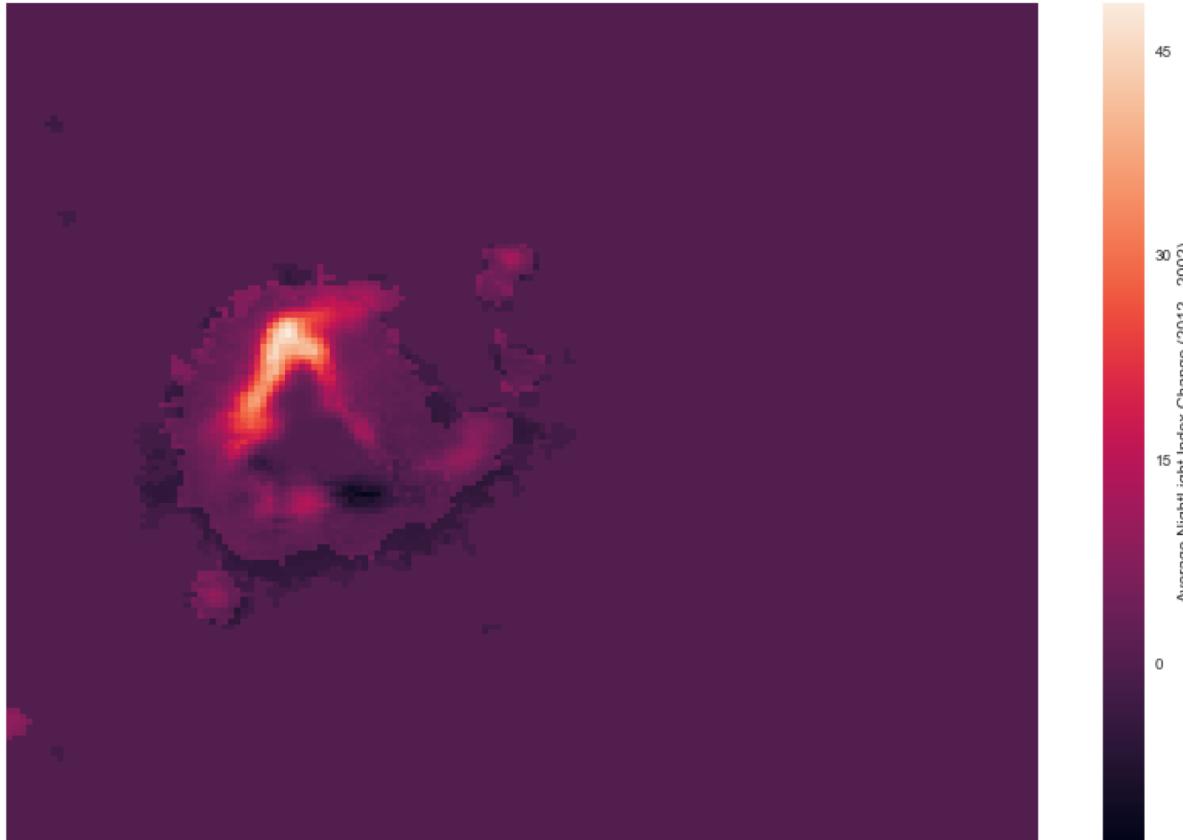
NDVI (Vegetation Index) STD by Neighborhood



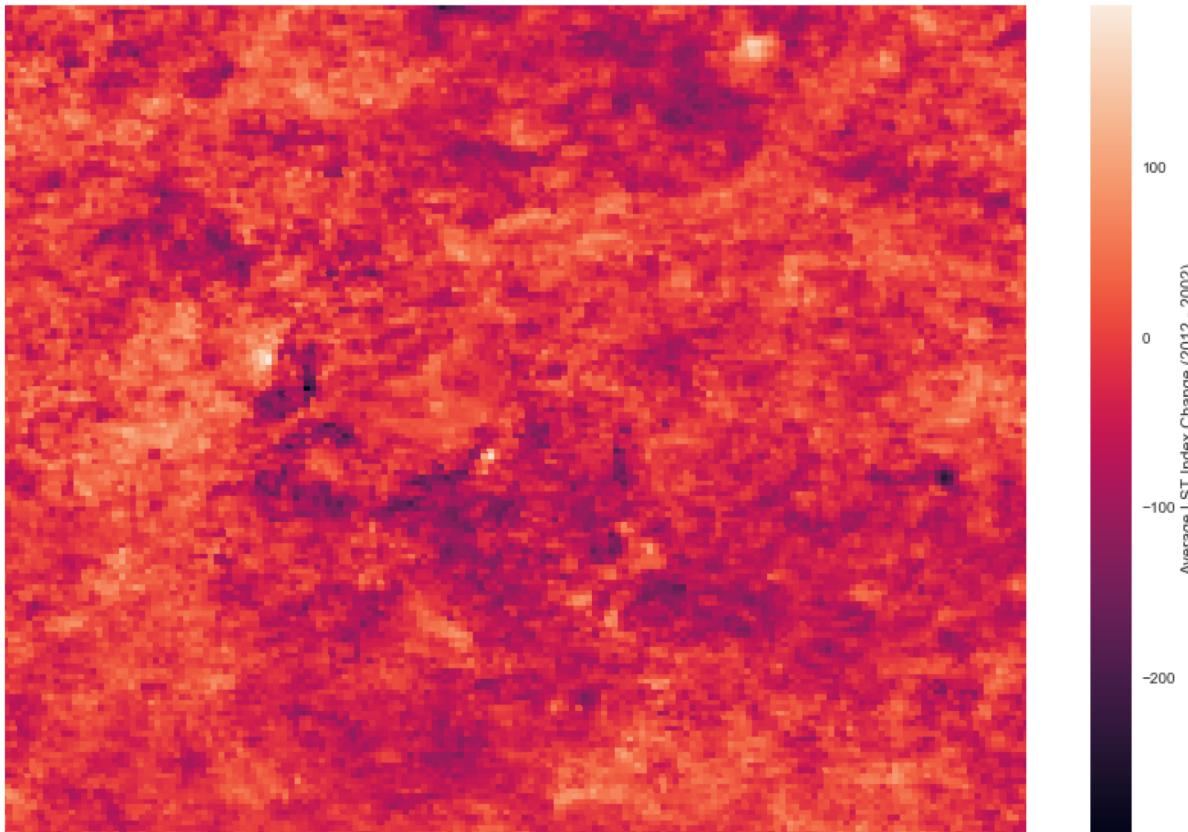
Section 3. Change Over Time: NDVI (Vegetation)



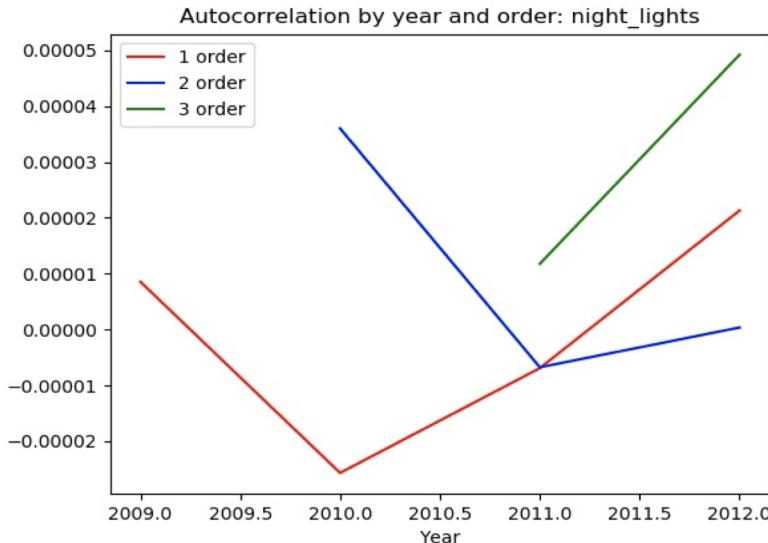
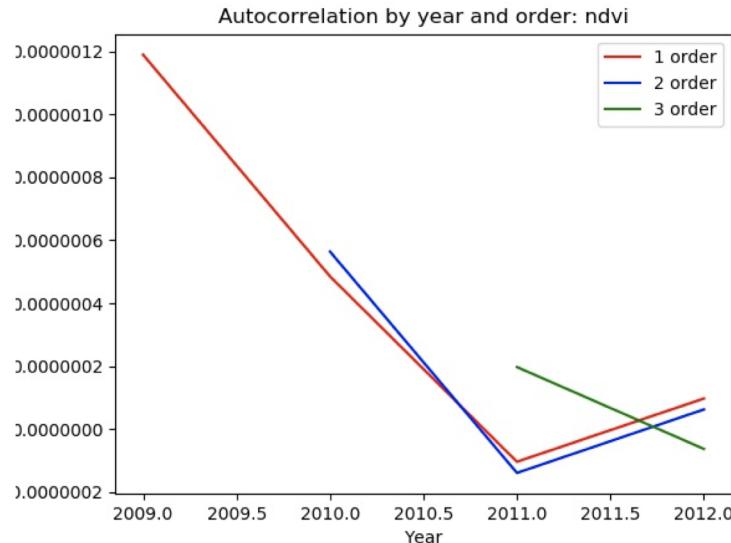
Change Over Time: Night Lights Index



Change Over Time: Land Surface Temperature



Section 4. Autocorrelation through time



Challenges

Challenges

- Unstructured data: getting google earth data into useable format
 - SatData class structure
- Annoyance of continually booting up and interfacing with VM instances
 - Bash scripts
- Memory issues
 - Larger memory
 - Processing sections separately

Questions?
