Using PostgreSQL and PostGIS for Climate Resilient Agriculture

Asim Rama Praveen, Sameer Mannava

Outline

- Who are we?
- What is PoCRA?
- Climate impact on agriculture
- Water budget framework
- Water budget based planning for climate resilience
- A tool for field staff - representative sample of farm plots
- Conclusion

Who are we?

- A team of field staff, students, consultants, faculty @ IIT Bombay
- Employed by Prof. Milind Sohoni, principal investigator
- Currently deputed on PoCRA



What is PoCRA?

World Bank sponsored **P**roject **o**n **C**limate **R**esilient **A**griculture $600 million (Rs. 4500 Cr)

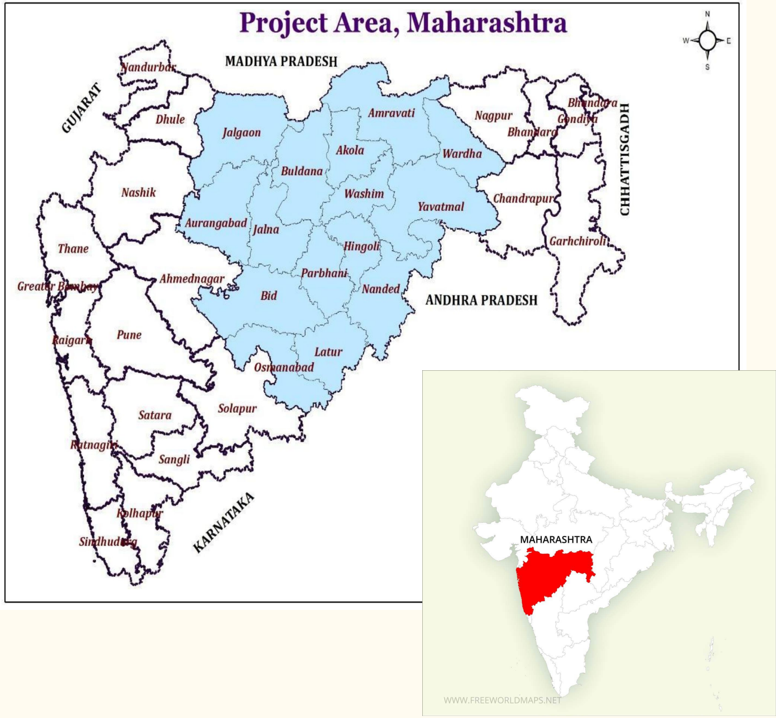
Series of 5 MoUs between IIT Bombay and Department of Agriculture, Maharashtra state

https://www.cse.iitb.ac.in/~pocra/

Time frame: 2018 to 2024

Project objective: improve climate resilience of smallholder farmers

Region: 5000+ villages in 15 districts



Impact of climate change

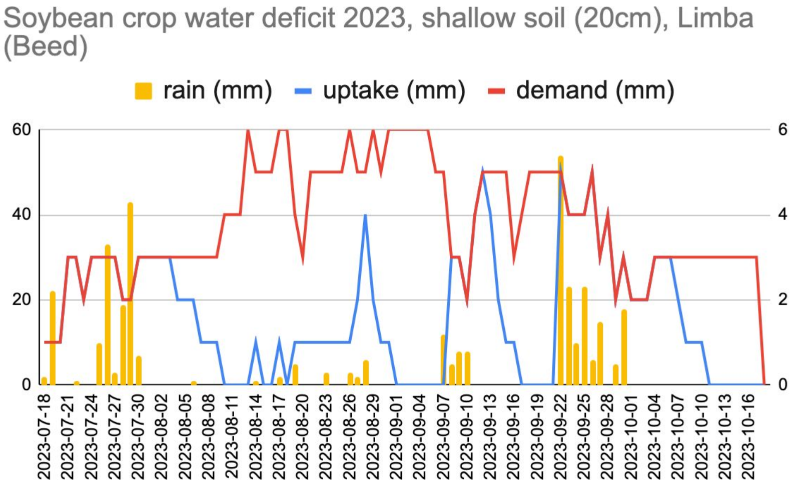
- Long dry spells during monsoon
  - 25+ days in Aug 2023
  - Crop water deficit
- Wet spells
  - 124mm in 1 hour (Yawatmal on 22 July 2023)
  - Water logging & erosion
- Hivargavhan farmer: (70% loss)
  - cotton yield down from 6 to 2 quintal/acre
- Tembha farmer: (40% loss)
  - cotton yield down from 8 to 5 quintal/acre
- The most impacted: small farmers (<2ha)
  - who depend solely on agriculture

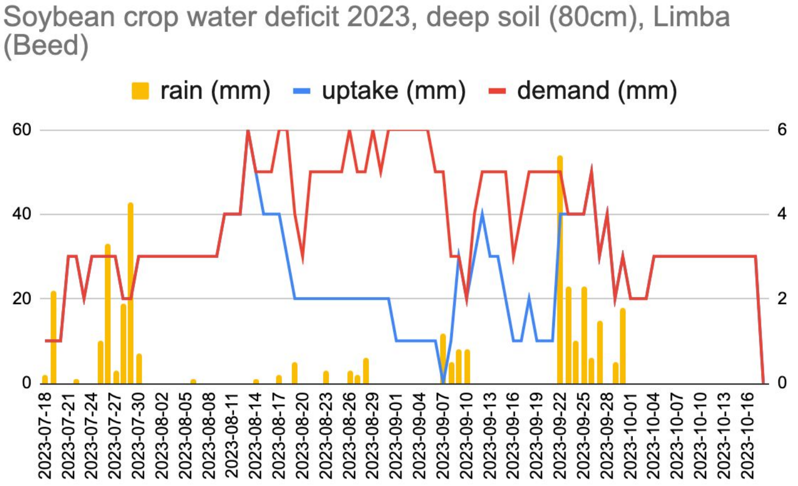
Hivargavhan, (Beed) Oct 2023

Tembha, (Wardha) Sept 2023

Impact of climate change

- Same village different farms - vastly different impact
- How to quantify vulnerability?
- How to measure climate resilience?
- How to improve it?





Current solution

indianexpress.com

November 30, 2023

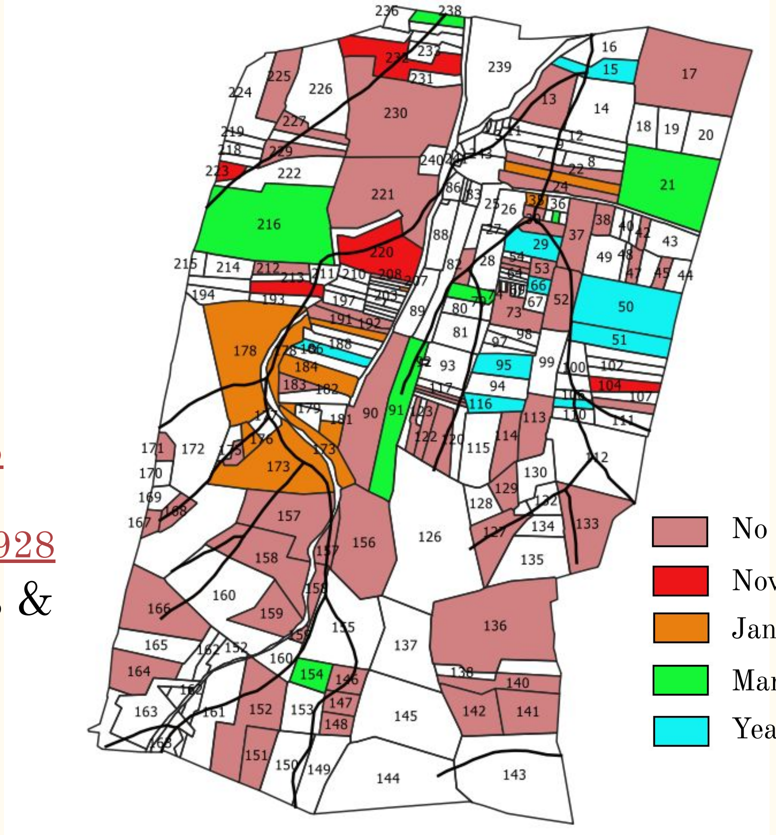
10k crore approved to compensate farmers in June 2022-Sept 2023, Maharashtra cabinet told

Alok Deshpande

- Why 10,000 crore? - aged methodology
- Lived reality remains grim

Climate resilience: village as a unit

- Baseline: capture lived reality
  - Median yield (quintal/acre)
  - Access to irrigation
  - Availability of water
- Biophysical vulnerability
  - Crop water deficit and their locations
  - Supply demand allocation framework
  - Water Policy https://doi.org/10.2166/wp.2023.036
  - CACM https://dl.acm.org/doi/10.1145/3554928
- Enable DoA to target interventions & advisories
- Make it work for 20000 villages
  - PostgreSQL, PostGIS, Geoserver



Cadastre wise water availability, Hivargvhan, (dist. Beed)

No source
November
January
March
Year round

Climate resilience: village as a unit

- Baseline: capture lived reality
  - Median yield (quintal/acre)
  - Access to irrigation
  - Availability of water
- Biophysical vulnerability
  - Crop water deficit and their locations
  - Supply demand allocation framework
  - Water Policy https://doi.org/10.2166/wp.2023.036
  - CACM https://dl.acm.org/doi/10.1145/3554928
- Enable DoA to target interventions & advisories

Boots on the ground

Water budget model, developed by IIT Bombay

Water budget frontend

Climate resilience: village as a unit

- Baseline: capture lived reality
  - Median yield (quintal/acre)
  - Access
  - Availabi
- Biophysica
  - Crop wa
  - Supply
  - Water Policy
    https://doi.org/10.2166/wp.2023.036
  - CACM
    https://dl.acm.org/doi/10.1145/3554928
- Enable DoA to target interventions & advisories

Field work, DoA

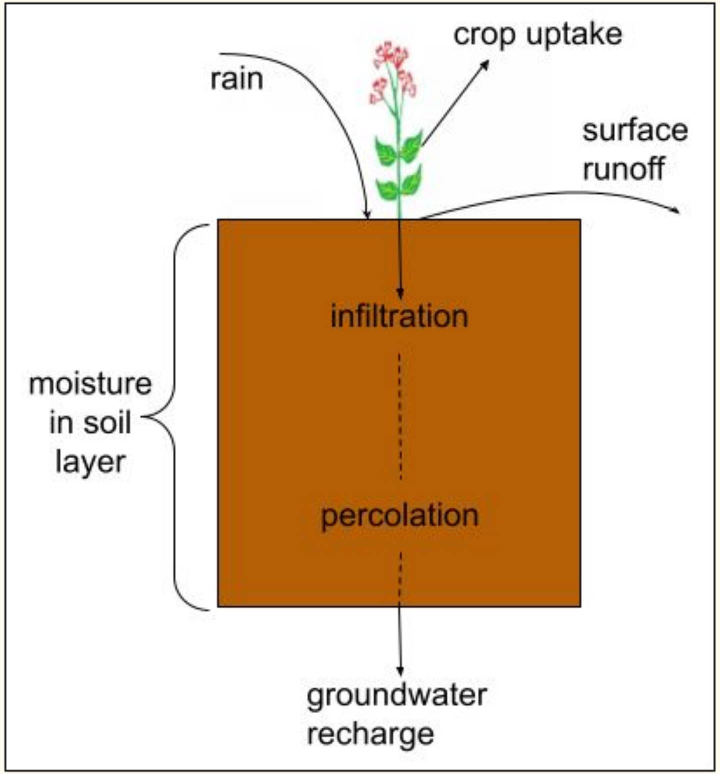
model, developed by IIT Bombay

Water budget frontend

All of this is backed by PostgreSQL and PostGIS

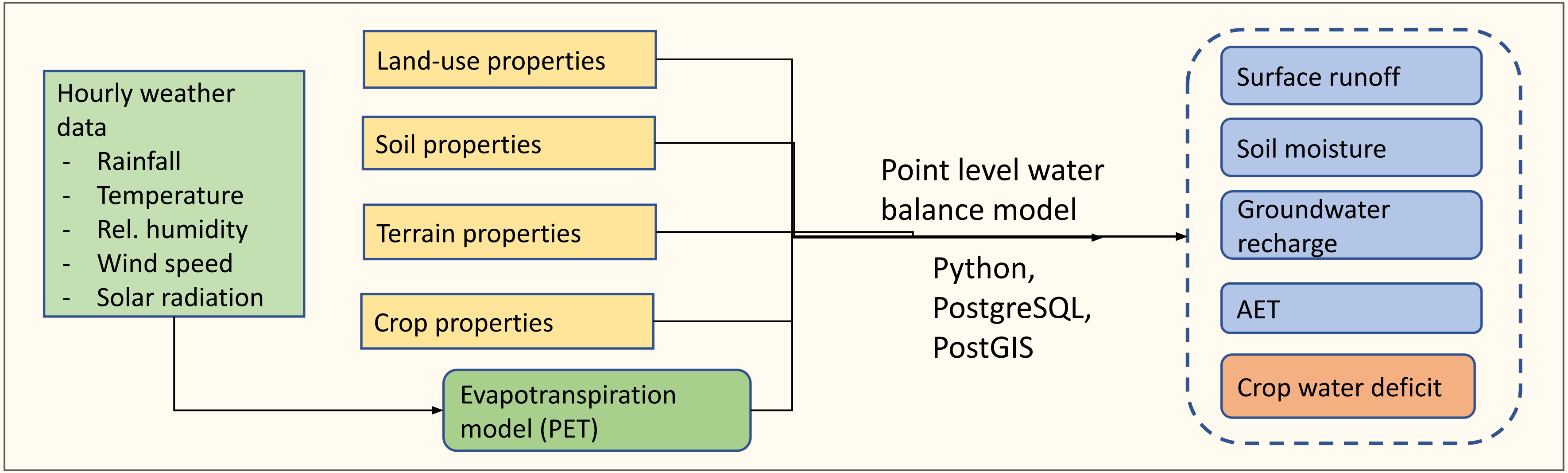
Water Budget Model

Water budget model: computations at a point



Distribution of rainfall into stocks

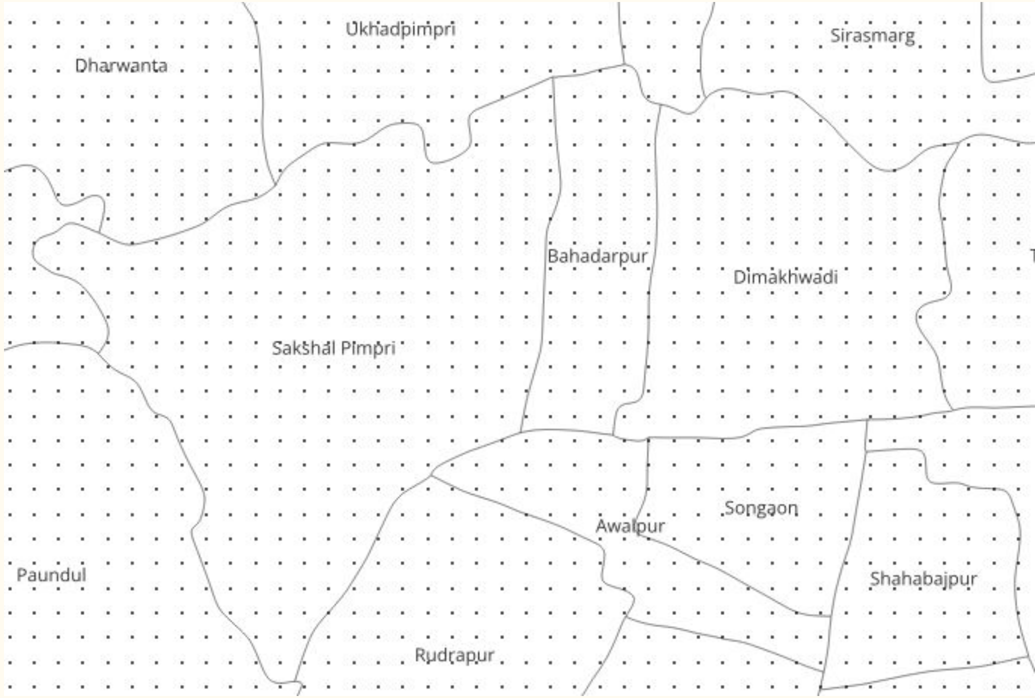
Water budget model: computations at a point



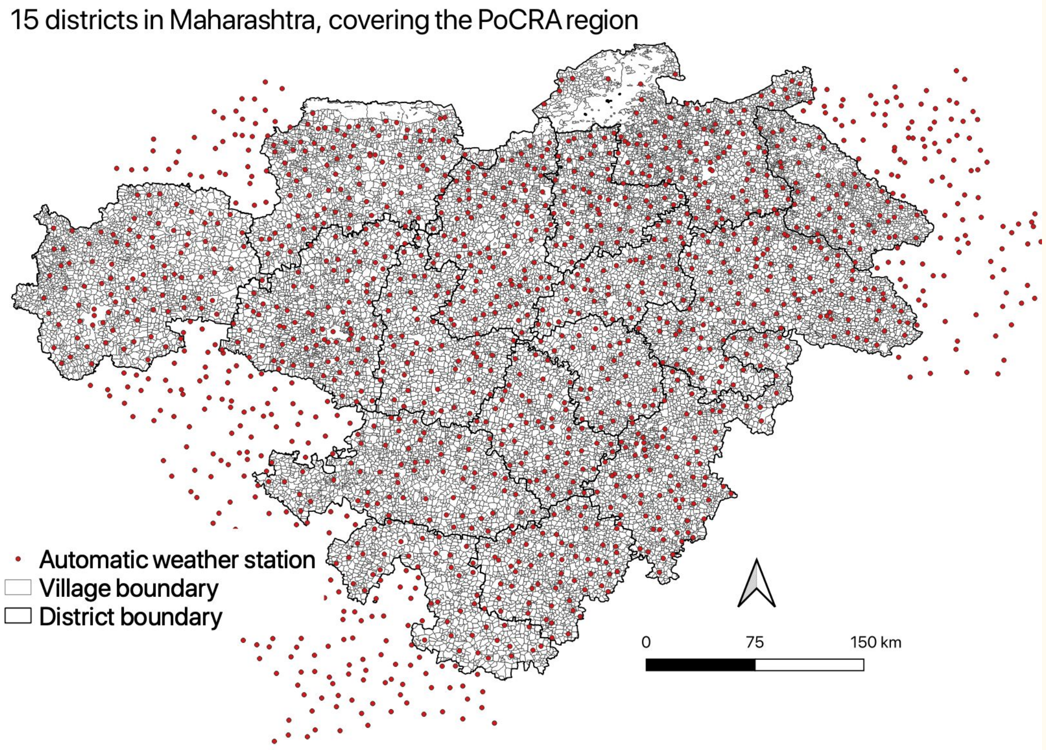
- Distribution of rainfall into stocks
- Vulnerability in terms of crop water deficit
- Dynamic: hourly/daily output during monsoon
- Validated in multiple villages

Water budget model: grid of points

- Points table: a district is divided into square grid of size 200m.
- Input parameters are added as columns to the grid points table
- Output table: one new record per crop, per day, per point
- 15 districts → 129,000 sq. km.
- 3,225,000 points
- 150 days * 32 crops = 15.48B output records per monsoon



Water budget inputs: hourly weather



Skymet API
@5am daily

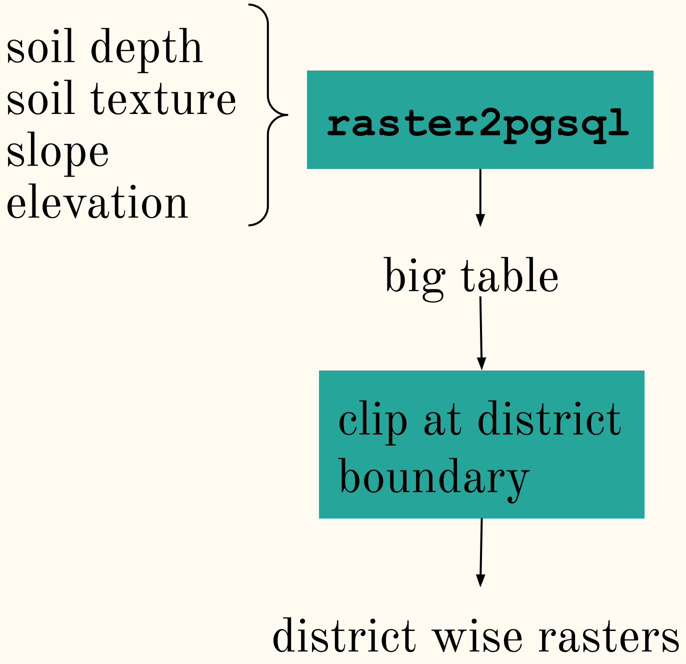
weather_stations
- geom
- staiton_id

hourly_weather
- station_id
- date
- hour
- rain
- temperature

Water budget inputs: hourly weather

- Weather stations may go down
- Model doesn’t tolerate missing data
- Identify missing data slots (station_id, date, hour)
- Replace null values with data from the nearest weather station

Water budget inputs: raster data




```sql
insert into district_wise_raster
 select
  st_clip(
    big.rast,
    st_buffer(small.geom, 100),
    -32768
  ) as rast
 from
  bigtable as big
 inner join
  districts as small
 on
  st_intersects(
    big.rast,
    st_buffer(small.geom, 100))
 and
  small.district_name = '...');
```

Attaching raster data to grid points

- Points table → 200m squares
- Raster table pixel size → 30m
- Intersection of a point square with raster tile → multiple raster values
- Value at the point:

  `(st_summarystats()).mean`

```
select
 point.id,
  (st_summarystats(
    st_union(
     st_clip(
      soil.rast,
      st_expand(point.geom, 100)))
   )
  ).mean as soil_depth
from
 {district}_soildepth as soil
inner join
 {district}_points as point
on
 st_expand(point.geom, 100)
  && s.rast
group by
 p.id
```

Water budget inputs: vector data

- Administrative boundaries
  - district, taluka, village
- Land use land cover (LULC)
  - agriculture (kharif, rabi), forest, habitation, ...
- st_containsproperly(): grid points within a land use class

```
select
  point.id
  Lulc.class_id as landuse_class
from
  {district}_points as point
join
  {district}_lulc as lulc
on
  st_containsproperly(
   lulc.geom,
   point.geom
  )
```

Assign nearest weather station to each village

- Villages: polygon geometry
- Weather stations: point geometry
- Lateral join, order by distance (PostGIS **<->** operator), limit 1

```
select
  v.vincode,
  nw.station_id as nearest_station,
  nw.distance
from
  villages v
cross join lateral
  (select
     station_id,
     st_distance(
       w1.geom,
       v.geom
     ) as distance
   from
     Weather_stations w1
   order by
     v.geom <-> w1.geom
   limit 1) as nw
```

Water budget output: results table

District wise results table
- point_id
- date
- crop
- uptake
- deficit
- soil_moisture
- runoff
- groundwater_recharge

New records inserted daily

Water budget daily cadence

Fetch hourly weather, smoothen missing data

Trigger parallel district runs

District run:

```
for each weather station in district
 read hourly weather
 for each point
  for each crop
   advance state: compute new stocks
   hourly to daily accumulation
   insert new record into output table
```

Water Budget: Demand Side

Planning for rabi (post monsoon season)

- Village wise cropping pattern (from DoA)
  - <crop, crop_area> pairs
- Exact crop uptake for the village
  - village_crop_uptake = avg((corp_area/village_area) * uptake)
- Volumetric water budget
  - (point wise output in mm) * (village agricultural area)

Planning for rabi (post monsoon season)

- Water available in Hivargavhan (536 ha) on 31 October 2023

Rainfall	Crop uptake	Runoff	Impounded runoff	Soil moisture	Groundwater
3769 TCM	911 TCM	1515 TCM	2 TCM	70 TCM	394 TCM

- Gram water requirement: 300 mm
- Feasible rabi area under gram: 156 ha (29% agricultural land)

Planning for rabi (post monsoon season)

- Water available in Hivargavhan (536 ha) on 31 October 2023

Rainfall	Groundwater
3769 TCM	394 TCM

Sustainable cropping pattern for the village based on water budget

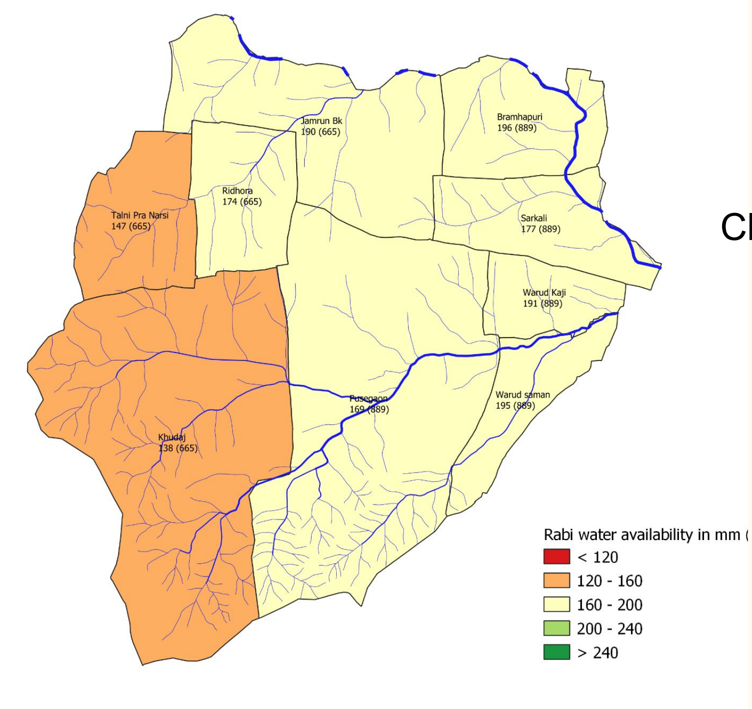
- Gram water re
- Feasible rabi area under gram: 154 ha (28% agricultural land)

We tried this exercise in12 villages

Takeaways for village community

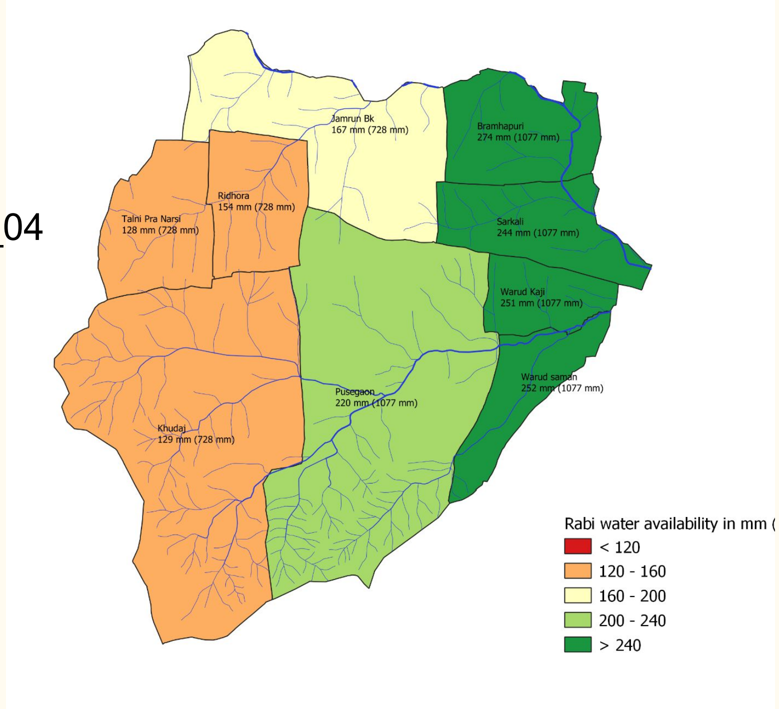
- Knobs for sustainable rabi cropping pattern:
  - Area under crop
  - Choice of crop
- Long term (before next monsoon)
  - Structures to impound surface runoff

Yearly variation in rabi water availability



Year 2018

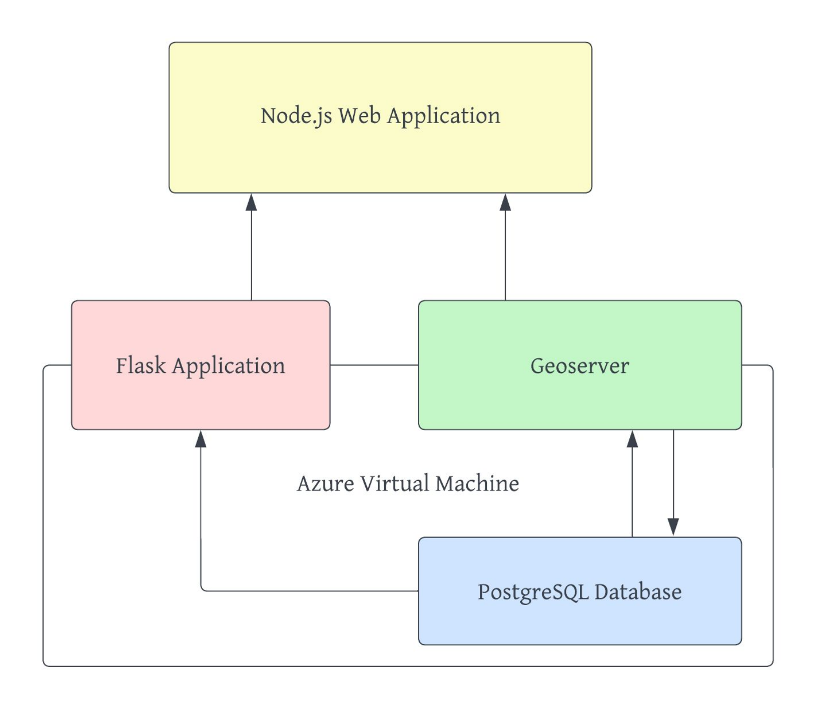
Cluster: 512_ppg-1_04
Taluka: Hingoli
District: Hingoli



Year 2022

Water Budget Reporting

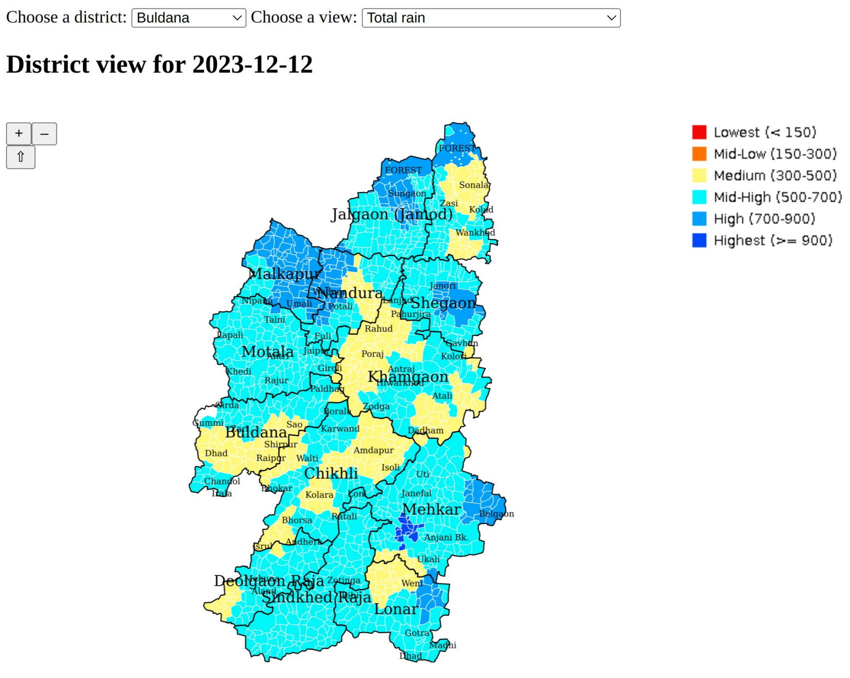
Water Budget Reporting Stack



- Spatial and temporal aggregation of daily water budget results
- End users:
  - DoA planners at district and taluka
  - Village community, village level DoA staff
- Geoserver to host maps
- Flask to run PostGIS queries (e.g. village wise soybean deficit on 07 Oct)
- UDFs to query water budget that accept district code, taluka code, crop, date, etc.

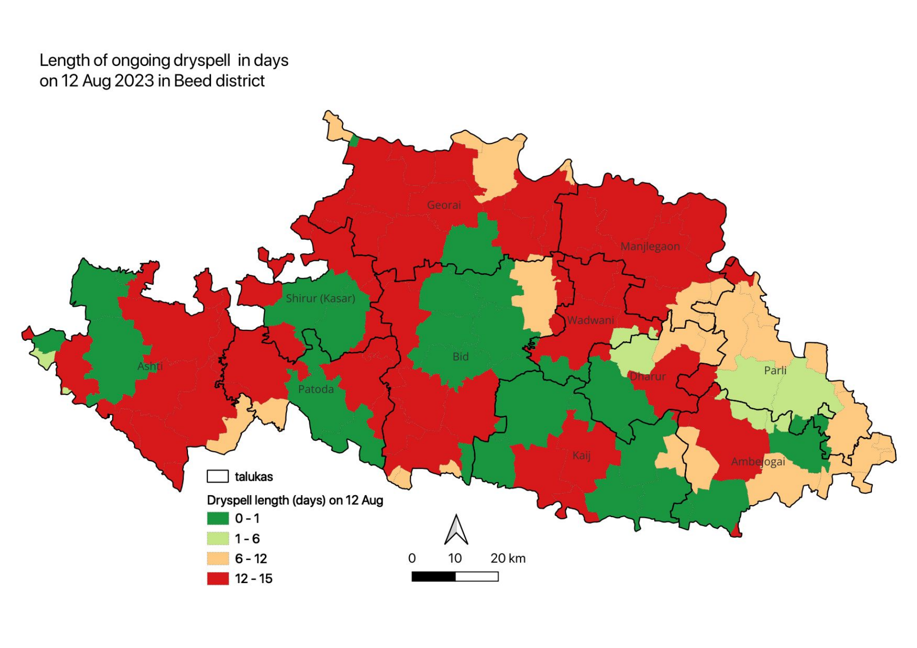
The District View

1) Total rainfall
2) Rainfall in the last 7 days
3) Length of the ongoing dry spell
4) Forecasted rainfall in the next 5 days



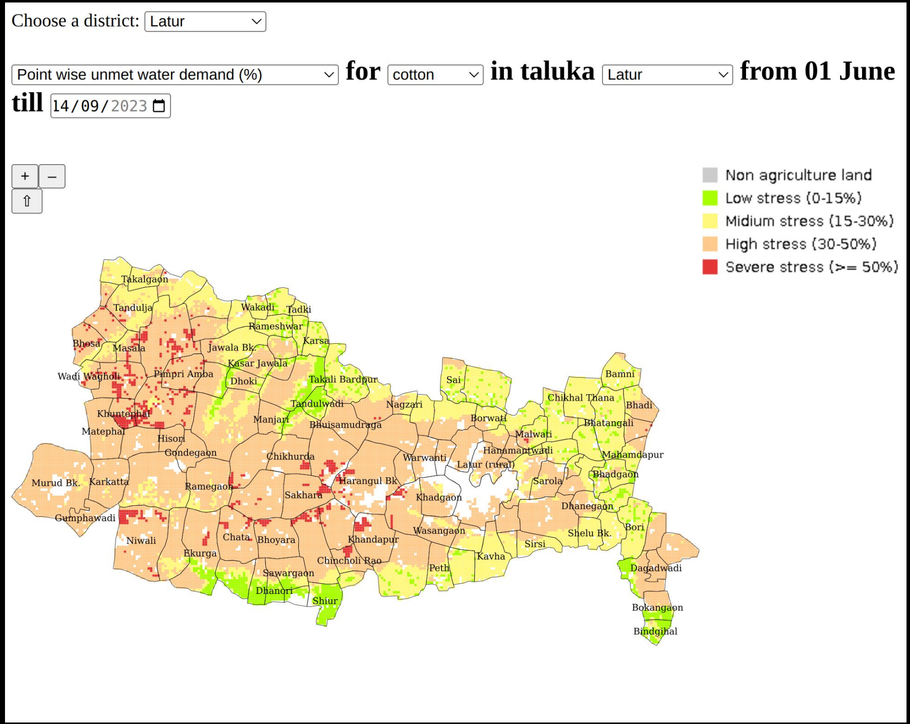
The District View

Length of the ongoing dry spell for Beed district on 12 Aug 2023



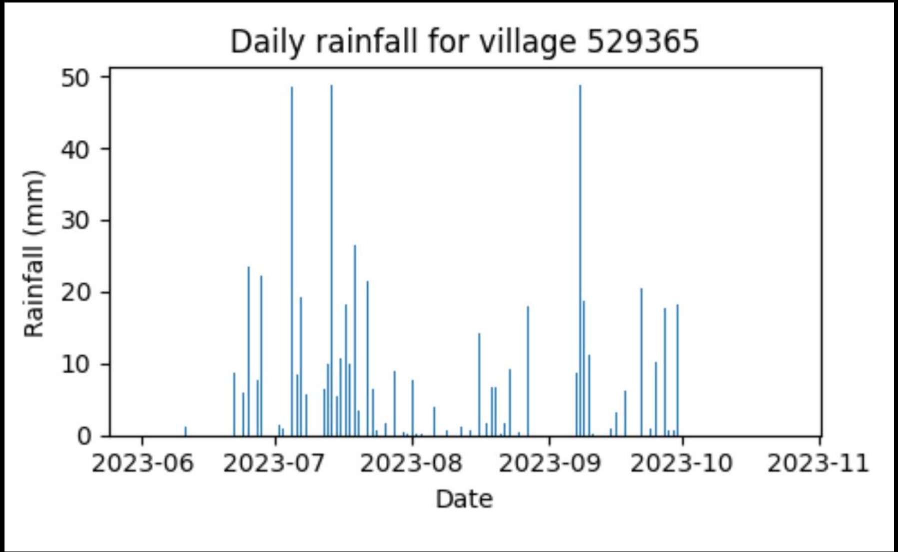
The Taluka View

1) Deficit percent
2) Available soil moisture
3) Need for protective irrigation



The Village View

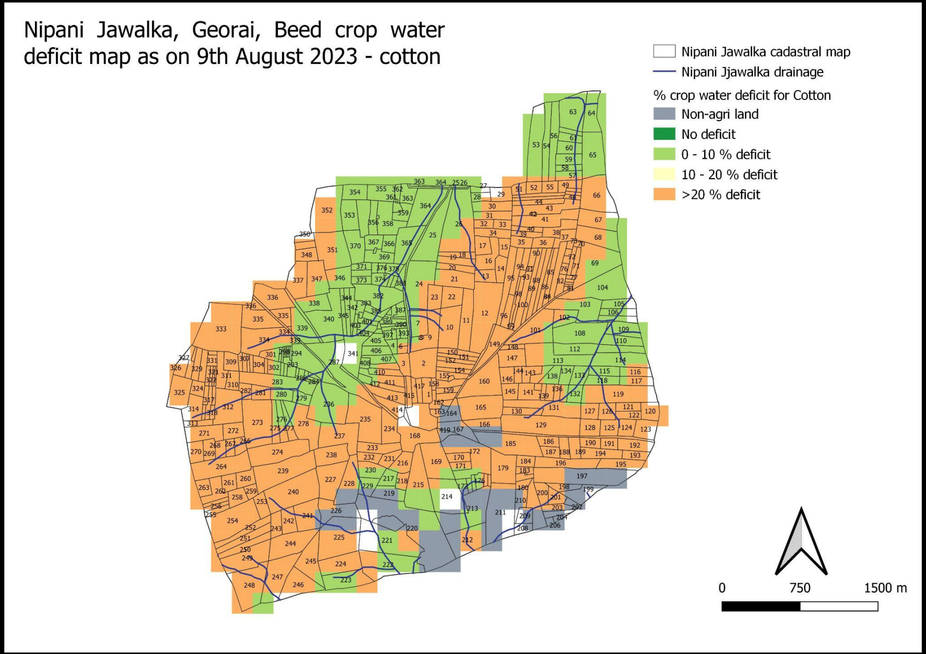
- A plot of daily rainfall
- Cumulative water budget results



Rainfall (mm) in the last 7 days	87.75
Total rainfall (mm) since the start of monsoon	499.00
Average unmet water demand (%)	22
Total groundwater recharge (mm)	84
Total surface water runoff (mm)	208

The Village View

Point wise deficit for cotton, village Nipani Jawalka, on 9 Aug 2023



Sampling Tool

Conducting surveys and validating our models

- Project spread across 20000 villages, 44000 villages in Maharashtra
- How many farmers do we survey in any given village?
- How to pick out of the numerous (~500) farm plots?
- How to achieve a good sample space? Four criteria to consider:
  - Need varying types of soil
  - Need varying levels of elevation
  - Need to survey stream-side and land-locked farms
  - Need a good geographical spread

Soil depth partition

- {district}_soildepth - district wide raster with soil depth data
- st_clip() clips the raster at the village boundaries
- st_reclass() classifies into predefined ranges
- st_dumpaspolygons() converts the raster into vector geometry

```
select
    vincode,
    (st_dumpaspolygons(
        st_reclass(
            st_clip(rast, geom),
            1, ‘[0-25]:1, (25-60]:2,
            (60-9999]:3’, ‘4BUI’, 0
        )
    )).*
from
    {district}_soildepth
join
    village
on
    st_intersects(rast, geom)
```

Elevation partition

- Elevation varies widely from village to village
- Predefined classes not feasible
- st_pixelaspoints(): Divide the village into pixels
- row_number(): Order the pixels by elevation
- st_summarystats(): Get the total number of pixels

```
elevation_pixels as (
    select
        vincode, (st_pixelaspoints(rast)).*
    from
        elevation_raster
),
elevation_ranked as (
    select
        vincode, val,
        row_number() over (order by val, x, y) as
rownum
    from
        elevation_pixels
),
elevation_summary as (
    select
        vincode, (st_summarystats(rast)).*
    from
        elevation_raster
),
```

Elevation partition

- Finally, divide into 33% and 67% percentile classes using a case statement

```
elevation_classified as (
    select
        r.vincode, r.val, r.rownum,
        case
            when
                r.rownum <= s.count/3
            then
                1
            when
                rownum > s.count/3 and rownum <= 2*s.count/3
            then
                2
            else
                3
        end as elevation_class
    from
        elevation_ranked r
    join
        elevation_summary s
    using
        (vincode)
)
```

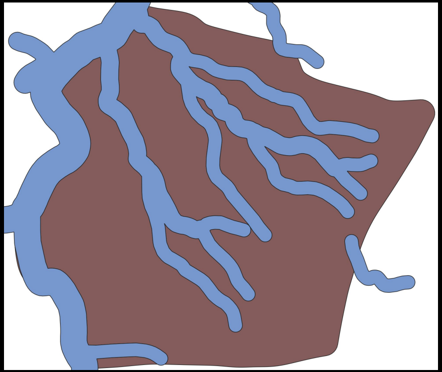
Stream proximity partition

- staticdata.“River” - Line geometries with magnitude indicating size of the stream
- st_buffer(): create a polygon geometry indicating ‘stream-fed’ farms
- Next, we take its complement using st_difference() to represent the ‘land-locked’ farms

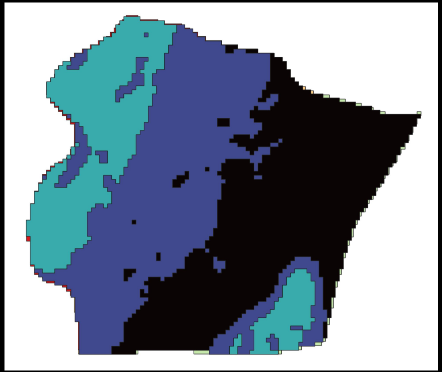
```
buffered_streams as (
    select
        st_union(st_buffer(geom::geography,
magnitude)::geometry) as streamfed
    from
        staticdata.”River”
),
select
    streamfed,
    st_difference(v.geom, streamfed) as
landlocked
from
    buffered_streams
join
    village
on
    true
```

Division of the village into distinct biophysical zones

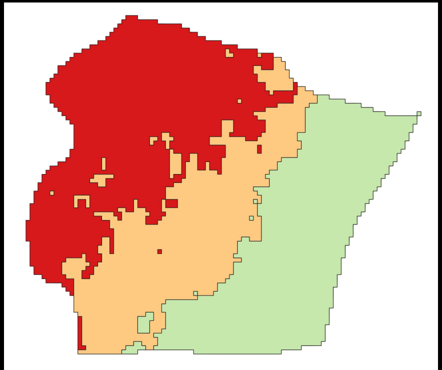
- Intersect the partitions created to form biophysical zones (3*3*2 = 18 types of zones possible)
- A new table is created with 5 columns:
  - farmplot id
  - geometry
  - soil depth class
  - elevation class
  - stream proximity



Stream proximity partition



Soil depth partition



Elevation partition

Division of the village into distinct biophysical zones

- random() is employed here so that the tool can be rerun if we desire a slightly different collection of farmplots

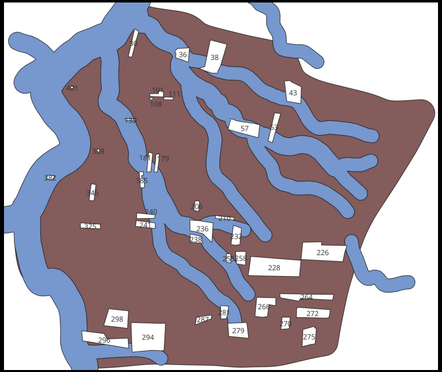
```
select
    pin,
    geom,
    eleclass,
    depthval,
    stream_region,
    row_number() over (
        partition by eleclass,
        depthval, stream_region
        order by random()
    ) as index
from
    biophysical_zones
```

How to achieve a good geographical spread?

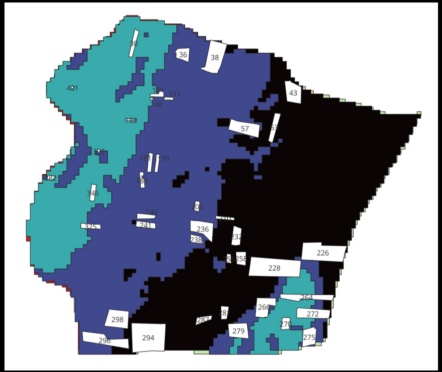
- `drop_all_touching_farms` drops touching farms using st_touches() to evaluate adjacency
- The following algorithm is run:

```
for zone in biophysical zones:
    for farmplot in zone if not dropped:
        drop_all_touching_farms(farmplot)
```

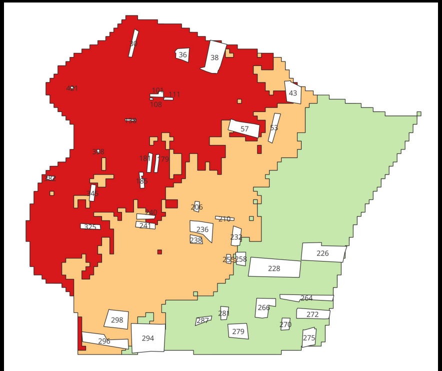
- The results overlaid on all the partitions ->



Stream proximity partition



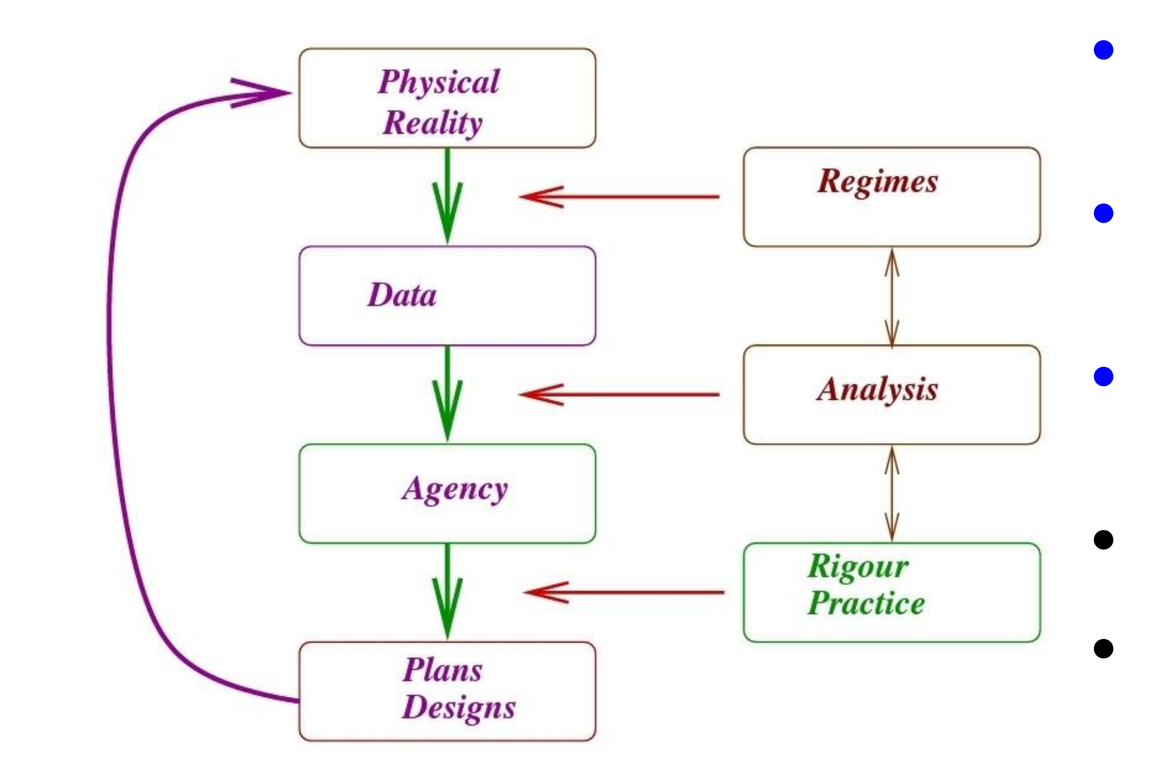
Soil depth partition



Elevation partition

In Summary

The infinite loop



- Generate data by capturing reality: baseline. Modernise the state methodology.
- The analysis of cause and effect, the parameters - create the authority and the will to act.
- The rigour and practice translates this will into concrete plans or designs.
- Thereby create space for AI, analytics
- And generate value. We are paid professionals!

Summary

PostgreSQL, PostGIS and geospatial community is

- enabling data driven decisions at village level
- helping DoA exercise its mandate in a sound scientific framework

We are also working on

- Generating accurate land parcel maps with Directorate of Land Records
- Flow based supply demand framework with Forest Dept. Himachal
- Drinking water and road network with CEO of Zilla Parishad, Ratnagiri district

Summary

PostgreSQL, PostGIS and geospatial community is

- enabling data driven decisions at village level
- helping DoA exercise its mandate in a sound scientific framework

IIT Bombay team is forever in gratitude, thank you!

Asim: asimrp@lavabit.com Sameer: sameer.mannava7@gmail.com