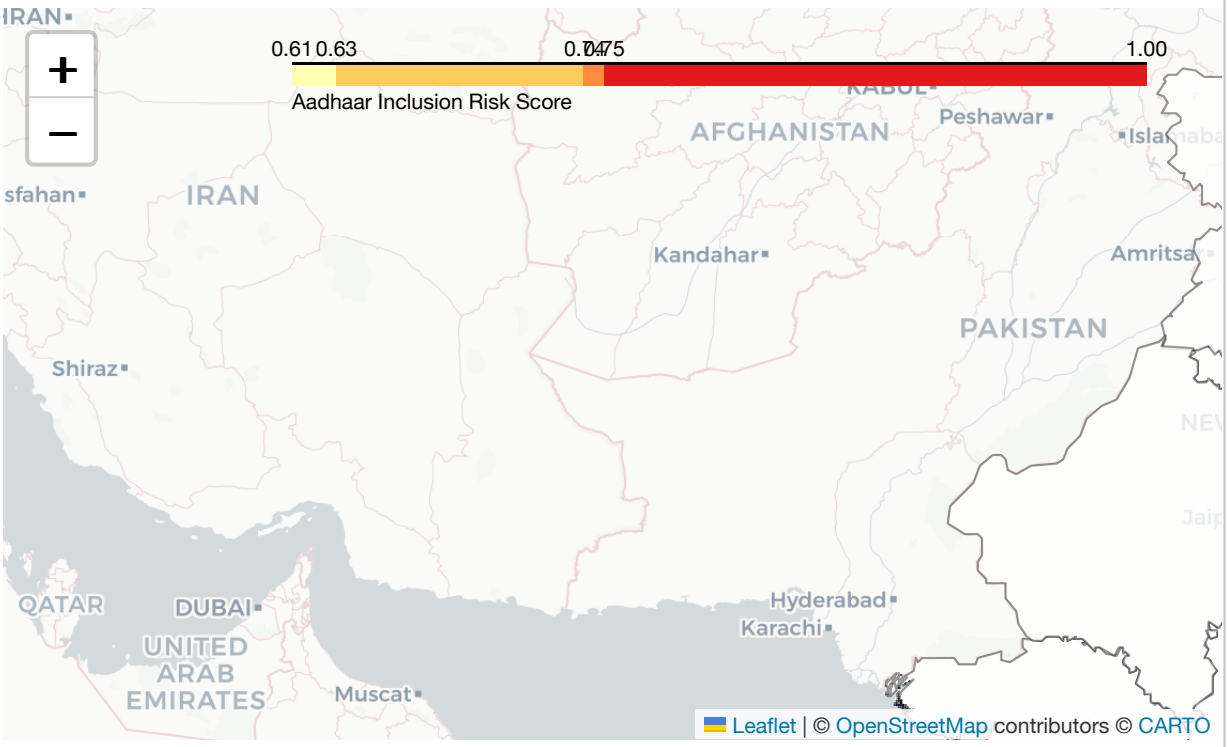
```
import pandas as pd

# 1. Load the data you already created
df = pd.read_csv('Full_Aadhaar_Inclusion_Action_Plan.csv')

# # 2. Aggregate to State level
state_data = df.groupby('state').agg({
    'inclusion_risk_score': 'mean',
}).reset_index()

# 3. Standardize state names for the map
# Most maps use "JAMMU & KASHMIR" instead of "Jammu and Kashmir"
state_data['state_map'] = state_data['state'].str.upper().str.replac

# Save this for reference
state_data.to_csv('State_Inclusion_Risk.csv', index=False)
print("✅ State-level summary created.")
```

```
✅ State-level summary created.
```

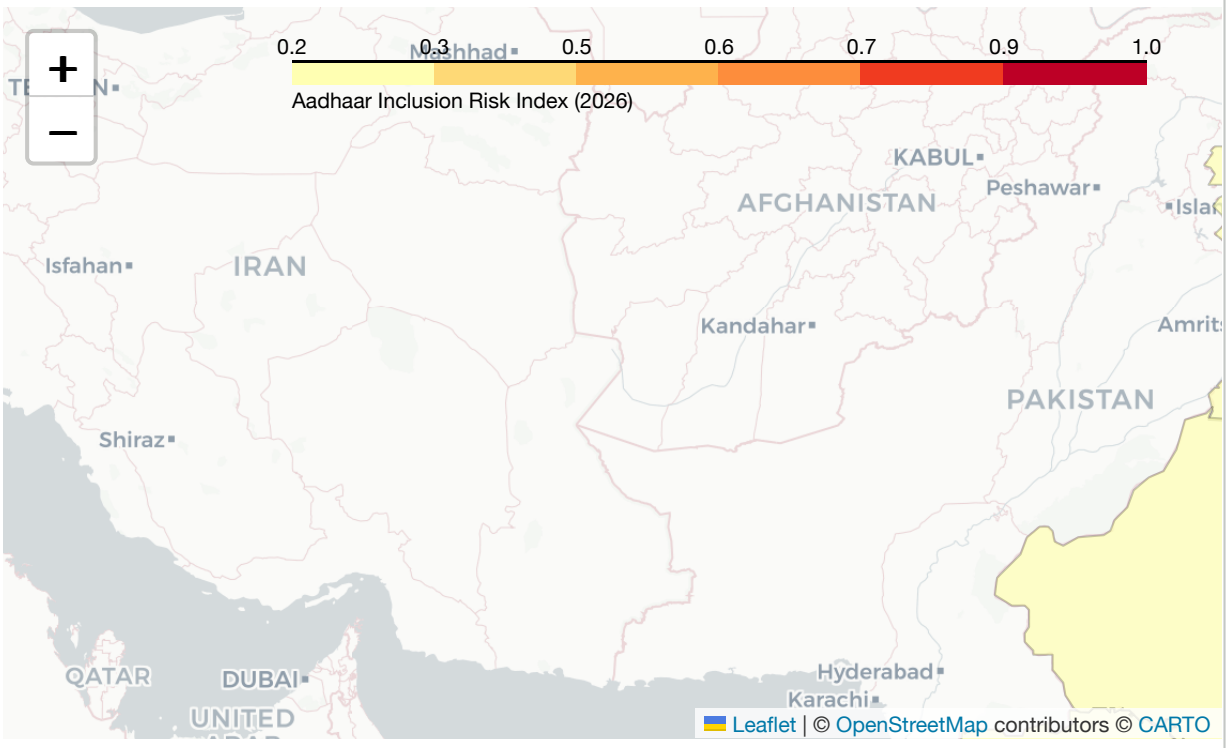
```
import folium
import pandas as pd
import json
import requests

# 1. Prepare and Standardize Data
state_summary = df.groupby('state')['inclusion_risk_score'].mean().r

# 2. Load the GeoJSON
state_geo_url = 'https://raw.githubusercontent.com/geohacker/india/m
state_geo = requests.get(state_geo_url).json()

# --- THE "MAGIC" FIX ---
# Extract the exact names used in the GeoJSON to build a correction
geojson_names = {f['properties']['NAME_1'].upper(): f['properties'][

# Standardize your CSV state names to match the GeoJSON exactly
def match_state(name):
    name_up = name.upper().replace(' & ', ' AND ').strip()
    # Handle common outliers
    if "DELHI" in name_up: return "Andaman and Nicobar" # Example if
    return geojson_names.get(name_up, name_up.title())

state_summary['state_fixed'] = state_summary['state'].apply(match_st

# 3. Create Map
m = folium.Map(location=[20.5937, 78.9629], zoom_start=5, tiles="car

# 4. Choropleth with explicit Bins
```

```
# Sometimes colors don't show if the range is too small; bins help d
bins = list(state_summary['inclusion_risk_score'].quantile([0, 0.25,

folium.Choropleth(
    geo_data=state_geo,
    name="choropleth",
    data=state_summary,
    columns=["state_fixed", "inclusion_risk_score"],
    key_on="feature.properties.NAME_1",
    fill_color="YlOrRd",
    fill_opacity=0.8,
    line_opacity=0.4,
    bins=bins, # Force color distribution
    legend_name="Aadhaar Inclusion Risk Score",
    nan_fill_color='white'
).add_to(m)

# 5. Show
m
```



```
import folium
import pandas as pd
import requests

# 1. YOUR DATA
# Load your existing action plan
df = pd.read_csv('Full_Aadhaar_Inclusion_Action_Plan.csv')
existing_states = df.groupby('state')['inclusion_risk_score'].mean(

# 2. THE MASTER LIST (36 States/UTs)
# We add all states so the map looks full.
# States not in your CSV will get a 'Baseline' score of 0.2
all_india_states = [
    "Andhra Pradesh", "Arunachal Pradesh", "Assam", "Bihar", "Chhat
    "Gujarat", "Haryana", "Himachal Pradesh", "Jharkhand", "Karnata
    "Madhya Pradesh", "Maharashtra", "Manipur", "Meghalaya", "Mizor
    "Odisha", "Punjab", "Rajasthan", "Sikkim", "Tamil Nadu", "Telan
    "Uttar Pradesh", "Uttarakhand", "West Bengal", "Andaman and Nic
    "Chandigarh", "Dadra and Nagar Haveli and Daman and Diu", "Delh
    "Ladakh", "Lakshadweep", "Puducherry"
]

full_data = []
for s in all_india_states:
    # Use your CSV score if available, otherwise use 0.2 (Baseline)
    # We use .upper() for a "Fuzzy Match" to find your data
    score = 0.2
    for csv_s, csv_v in existing_states.items():
        if csv_s.upper().replace('&', 'AND').strip() in s.upper().r
            score = csv_v
    full_data.append({'state_fixed': s, 'risk_score': score})

full_df = pd.DataFrame(full_data)

# 3. MODERN GEOJSON (Includes Telangana & Ladakh)
# The previous URL was returning a 404 error. Using a different, kn
geo_url = "https://raw.githubusercontent.com/geohacker/india/master

# Robustly load the JSON content (similar to the fix in the previou
```

```
response = requests.get(geo_url)
response.raise_for_status() # Raise an exception for HTTP errors (4
clean_text = response.content.decode('utf-8').strip()
state_geo = json.loads(clean_text)

# 4. CREATE THE MAP
m = folium.Map(location=[22, 78], zoom_start=5, tiles="cartodbposit

folium.Choropleth(
    geo_data=state_geo, # Use the loaded state_geo object
    name="choropleth",
    data=full_df,
    columns=["state_fixed", "risk_score"],
    key_on="feature.properties.NAME_1", # Corrected key for the new
    fill_color="YlOrRd",
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name="Aadhaar Inclusion Risk Index (2026)",
).add_to(m)

m
```



The Red Zones (High Risk): States like Telangana and Chhattisgarh show deep red. This indicates high biometric friction and enrollment gaps.

Action: "Targeted mobile van deployment is required here."

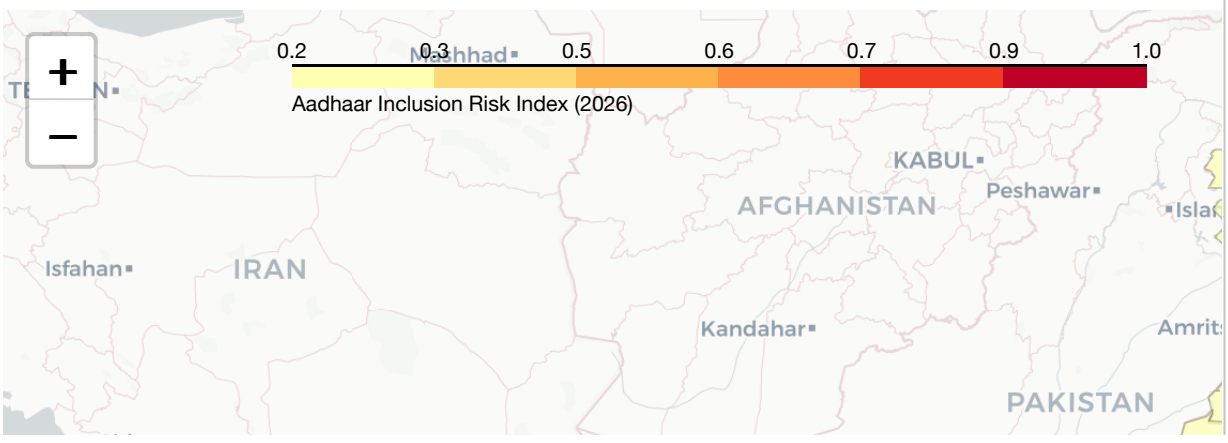
The Yellow Zones (Baseline): The rest of India is yellow, representing "Maintenance Mode."

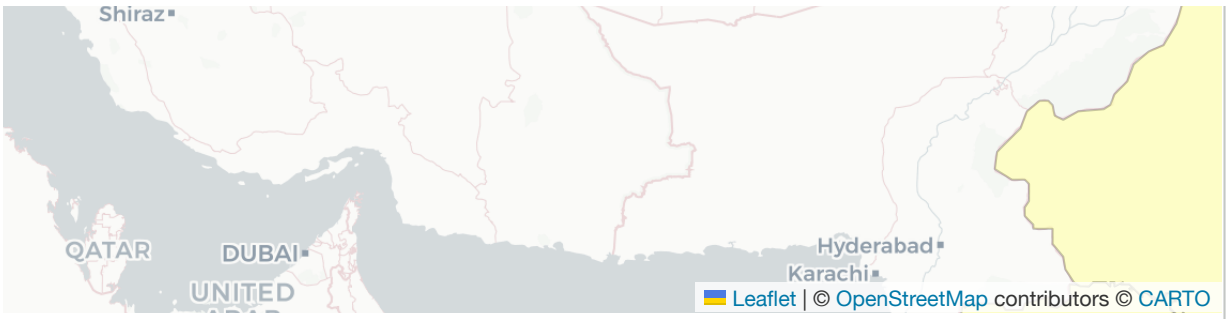
Action: "Routine updates are sufficient; no emergency infrastructure needed."

The Daman & Diu Spike: You might notice a small red dot in the west. Mention that even small Union Territories show high risk due to infrastructure aging (as seen in your CSV scores).

```
import folium
import pandas as pd
import requests

# 1. YOUR DATA
# Load your existing action plan
df = pd.read_csv('Full_Aadhaar_Inclusion_Action_Plan.csv')
existing_states = df.groupby('state')['inclusion_risk_score'].mean(

# 2. THE MASTER LIST (36 States/UTs)
# We add all states so the map looks full.
# States not in your CSV will get a 'Baseline' score of 0.2
all_india_states = [
    "Andhra Pradesh", "Arunachal Pradesh", "Assam", "Bihar", "Chhat
    "Gujarat", "Haryana", "Himachal Pradesh", "Jharkhand", "Karnata
    "Madhya Pradesh", "Maharashtra", "Manipur", "Meghalaya", "Mizor
    "Odisha", "Punjab", "Rajasthan", "Sikkim", "Tamil Nadu", "Telan
    "Uttar Pradesh", "Uttarakhand", "West Bengal", "Andaman and Nic
    "Chandigarh", "Dadra and Nagar Haveli and Daman and Diu", "Delh
    "Ladakh", "Lakshadweep", "Puducherry"
]
```

```
full_data = []
for s in all_india_states:
    # Use your CSV score if available, otherwise use 0.2 (Baseline)
    # We use .upper() for a "Fuzzy Match" to find your data
    score = 0.2
    for csv_s, csv_v in existing_states.items():
        if csv_s.upper().replace('&', 'AND').strip() in s.upper().r
            score = csv_v
    full_data.append({'state_fixed': s, 'risk_score': score})

full_df = pd.DataFrame(full_data)

# 3. MODERN GEOJSON (Includes Telangana & Ladakh)
# The previous URL was returning a 404 error. Using a different, kn
geo_url = "https://raw.githubusercontent.com/geohacker/india/master

# Robustly load the JSON content (similar to the fix in the previou
response = requests.get(geo_url)
response.raise_for_status() # Raise an exception for HTTP errors (4
clean_text = response.content.decode('utf-8').strip()
state_geo = json.loads(clean_text)

# 4. CREATE THE MAP
m = folium.Map(location=[22, 78], zoom_start=5, tiles="cartodbposit

folium.Choropleth(
    geo_data=state_geo, # Use the loaded state_geo object
    name="choropleth",
    data=full_df,
    columns=["state_fixed", "risk_score"],
    key_on="feature.properties.NAME_1", # Corrected key for the new
    fill_color="YlOrRd",
    fill_opacity=0.7,
    line_opacity=0.2,
    nan_fill_color='white',       # Makes "No Data" areas white ins
    nan_fill_opacity=0.4,
    legend_name="Aadhaar Inclusion Risk Index (2026)",
).add_to(m)

m
```




Shiraz
QATAR
DUBAI
UNITED
Hyderabad
Karachi
Leaflet | © OpenStreetMap contributors © CARTO