



Google  
Summer of Code



Google Summer of Code Proposal

# **Ecotourism: Data Package Containing Tourism Records and Endangered Wildlife Reports**


DEV GOEL  
18 March 2025

## Contents

1. Project Info .....	2
2. Bio .....	2
3. Contact Information .....	2
4. Affiliation .....	2
5. Schedule Conflicts .....	3
6. Project Mentors .....	3
7. Coding Plan and Methods .....	3
7.1. Data Acquisition and Preprocessing .....	3
7.2. Automated Data Refresh Pipeline .....	4
7.3. Geospatial Matching and Temporal Analysis .....	4
7.4. Dockerized Environment .....	4
7.5. Parallelized Data Processing .....	5
7.6. Automated CRAN Compliance Checks .....	5
7.7. Challenges and Contingency Plan .....	5
7.8. 8. Future-Proofing and Scalability .....	6
8. Timeline .....	6
9. Management of Coding Project .....	8
9.1. Testing Strategy .....	8
9.2. Weekly Blog Updates .....	8
9.3. Post-GSoC Contributions .....	9
9.4. Communication and Collaboration .....	9
10. Test Submissions .....	9
10.1. Easy Test: .....	9
10.2. Medium Test: .....	9
10.3. Hard Test: .....	10

---

## 1. Project Info

- **Project Title:** Ecotourism: Data Package Containing Tourism Records and Endangered Wildlife Reports
- **Short Title:** Ecotourism Data Package
- **Project Page URL:**  [Ecotourism Project Page](#)

## 2. Bio

I am Dev Goel, a B.Tech student at IIT BHU majoring in Pharmaceutical Engineering. My programming journey began in 7th grade as a self-taught enthusiast, building fun projects like a neural network-powered Flappy Bird game. Over the years, I have expanded my skills in Python and explored data science and machine learning.

My interest in R sparked during an exploratory project on pharmacodynamics and pharmacokinetics, where I analyzed medicine-related data. While exploring data science tools, I discovered R's rich ecosystem for statistical analysis and visualization, which drew me deeper into the language. When I came across the Ecotourism data package project in GSoC, I saw it as an excellent opportunity to combine my programming skills with geospatial data visualization and ecological analysis.

Beyond data science, I have hands-on experience in software and web development, equipping me with the technical versatility to handle data manipulation, visualization, and package development effectively. I am excited about the chance to collaborate with experienced professionals, learn from their expertise, and contribute meaningfully to the R community.

## 3. Contact Information

**Name:** Dev Goel

**Postal Address:** 729, Satish Dhawan Hostel, Indian Institute of Technology (BHU), Varanasi, India, 221005

**Phone Number:** [+91-8968998040](tel:+91-8968998040)

**Email(s):** [dgoel2099@gmail.com](mailto:dgoel2099@gmail.com), [dev.goel.phe23@itbhu.ac.in](mailto:dev.goel.phe23@itbhu.ac.in)

**Other Communication Channels:** Zoom, Google Meet, [Whatsapp](#)

## 4. Affiliation

**Institution:** Indian Institute of Technology (Banaras Hindu University) [IIT (BHU)], Varanasi, India

**Program:** B.Tech. in Pharmaceutical Engineering and Technology

**Stage of Completion:** 2nd Year, Expected Graduation: 2027



---

Contact to verify: Dr. Ashish Kumar Agrawal (Email: [ashish.phe@iitbhu.ac.in](mailto:ashish.phe@iitbhu.ac.in))

## 5. Schedule Conflicts

I will be fully available throughout the summer, as my college will be on summer vacation from May 10 to July 10. During this period, I will be able to dedicate 30–35 hours per week to the project. However, I will be traveling home at the start of the summer and back to college at the end, which will make me unavailable for 3–4 days during each of these transitions. To compensate for this lost time, I plan to work extra hours during the remaining weeks of the summer.

If the project extends beyond the summer, I will still be able to contribute 25 hours per week alongside my college coursework. I can dedicate approximately 3 hours on weekdays ( $3 \times 5$ ) and 5 hours on weekends ( $2 \times 5$ ). This ensures that I will continue to meet project milestones and contribute consistently, even after the summer coding period.

## 6. Project Mentors

Evaluating Mentor: Dianne Cook (Email: [dicook@monash.edu](mailto:dicook@monash.edu))

Co-Mentor: Lyn Cook (Email: [l.cook@uq.edu.au](mailto:l.cook@uq.edu.au))

I have been in touch with the mentors via email. They have provided valuable feedback on my project tests and have been very supportive throughout the process. I am looking forward to working with them on this project.

## 7. Coding Plan and Methods

In order to successfully complete the Ecotourism data package, I will follow a structured and methodical coding plan. This will include data acquisition, processing, geospatial matching, testing, and visualization. Additionally, I will incorporate advanced features such as automated data refresh, Dockerized environment, and parallelized processing to make the package more robust, efficient, and scalable.

### 7.1. Data Acquisition and Preprocessing

The project will begin by acquiring and preprocessing ecotourism and wildlife records from multiple sources:

- **GBIF API:** To fetch wildlife occurrence data.
- **eBird API:** For bird sighting records.
- **OSM API:** To obtain spatial data for mapping and visualization.

These sources may change along with the project requirements, and I will adapt the data acquisition process accordingly.

---

### Preprocessing Tasks:

- Cleaning and standardizing timestamps and coordinates.
- Removing duplicates and handling missing values.
- Performing initial exploratory data analysis (EDA) to identify patterns.

## 7.2. Automated Data Refresh Pipeline

To keep the data package consistently updated with the latest records, I will implement an automated data refresh pipeline using GitHub Actions.

### Feature:

- The pipeline will periodically fetch the latest data from GBIF, eBird, and OSM APIs.
- The refreshed data will be automatically committed to the GitHub repository.

### Technical Details:

- Use GitHub Actions with a scheduled CRON job that triggers weekly or monthly.
- Implement data validation checks to prevent erroneous data from being committed.

## 7.3. Geospatial Matching and Temporal Analysis

I will implement advanced geospatial matching techniques using the R packages:

- `sf` : For geospatial operations such as joins, intersections, and buffering.
- `osmdata` : To extract and match locations based on OpenStreetMap data.

### Technical Tasks:

- Perform spatial joins to link wildlife sightings with tourism hotspots.
- Apply temporal matching functions to combine records by date.
- Use distance thresholds to handle mismatched or imprecise coordinates.

## 7.4. Dockerized Environment

To ensure a consistent development environment, I will include Docker support for easy deployment and reproducibility.

### Feature:

- A Docker container will package the R environment with all dependencies pre-installed.
- This ensures that the package runs identically across different systems.

### Technical Details:

- Create a `Dockerfile` with R, required libraries, and datasets.
- Enable contributors to quickly deploy the package using:



- 
- `docker build -t ecotourism-r .`
  - `docker run -p 8787:8787 ecotourism-r`

## 7.5. Parallelized Data Processing

To improve the performance of geospatial joins and temporal operations, I will implement parallelized processing using:

- `future.apply` for parallelized operations in R.
- `parallel` for running multiple processes simultaneously.

### Benefits:

- Significantly reduces the processing time for large datasets.
- Enhances the overall efficiency of the package.

## 7.6. Automated CRAN Compliance Checks

To ensure the package is always CRAN-compliant, I will set up automated compliance checks using GitHub Actions.

### Feature:

- The workflow will validate CRAN compliance on every push.
- This ensures that the package passes all required checks before submission.

### Technical Details:

- Use `rcmdcheck` GitHub Action:
  - `uses: r-lib/actions/check-r-package@v2`
- Automatically validate documentation, dependencies, and code integrity.

## 7.7. Challenges and Contingency Plan

During the development of this project, I anticipate potential challenges, and I have devised a contingency plan to overcome them.

### Data Limitations:

- Some APIs may impose rate limits or temporary unavailability.
- **Solution:** Use caching and batching techniques to reduce API requests and prevent rate-limiting issues.

### Geospatial Challenges:

- Mismatched or inaccurate coordinates could cause errors in geospatial joins.
- **Solution:** Implement distance thresholds and apply fuzzy matching algorithms to handle imprecise data.

### Backup Plan:



- In case of API failures or unavailability, I will use local CSV datasets as a fallback to ensure continuous development without interruptions.

## 7.8. 8. Future-Proofing and Scalability

To make the package scalable and future-proof, I will:

- Use modular code architecture for easy maintenance and extension.
- Include detailed documentation and inline comments for future contributors.
- Make the codebase easily extensible for adding new data sources or functionality.

## 8. Timeline

Phase	Timeline	Deliverable	Details and Milestones
<b>Org Application Deadline</b>	11 Feb 2025	N/A	GSoC organization application deadline.
<b>Organizations Announced</b>	27 Feb 2025	N/A	Official GSoC organizations announced. Start contacting potential mentors.
<b>Pre GSoC Period</b>	8 Apr - 8 May 2025	N/A	Discuss project ideas with the mentors and build upon their earlier findings
<b>Bonding Period</b>	8 May – 2 June 2025	Environment setup, project planning	<ul style="list-style-type: none"> <li>• Set up development environment.</li> <li>• Identify and collect sample datasets from GBIF, eBird, and OSM.</li> <li>• Discuss project scope and expectations with mentors.</li> <li>• Create a detailed technical roadmap.</li> </ul>

<b>Coding Phase 1</b>	2 June – 18 July 2025	Data acquisition & cleaning	<ul style="list-style-type: none"> <li>• Implement data retrieval functions from GBIF, eBird, and OSM.</li> <li>• Clean and standardize data (timestamps, coordinates, missing values).</li> <li>• Perform exploratory data analysis (EDA) to identify patterns.</li> <li>• Push working prototype to GitHub with basic cleaning functions.</li> </ul>
<b>Midterm Evaluation</b>	18 July 2025	Midterm evaluation by mentors	Mentors evaluate project progress, functionality, and code quality.
<b>Coding Phase 2</b>	19 July – 8 August 2025	Spatial & temporal matching	<ul style="list-style-type: none"> <li>• Implement spatial matching functions using <code>sf</code> and <code>osmdata</code>.</li> <li>• Perform temporal matching with dynamic date ranges.</li> <li>• Optimize and test geospatial joins for efficiency.</li> <li>• Create basic plots and maps for verification.</li> </ul>
<b>Coding Phase 3</b>	9 August – 1 September 2025	Visualization & dashboard	<ul style="list-style-type: none"> <li>• Add animated heatmaps and time-series maps using <code>leaflet</code> and <code>plotly</code>.</li> <li>• Implement an interactive Shiny dashboard.</li> <li>• Perform optimizations and bug fixes.</li> <li>• Create vignette documentation with usage examples.</li> </ul>
<b>Final Evaluation Period</b>	1 – 8 September 2025	Final evaluation by mentors	<ul style="list-style-type: none"> <li>• Submit the complete project to mentors for review.</li> <li>• Ensure CRAN compliance.</li> </ul>



---

<b>Final Submission &amp; CRAN Release</b>	9 – 16 September 2025	Submit package to CRAN	<ul style="list-style-type: none"> <li>• Officially submit the package to CRAN.</li> <li>• Write the final project report summarizing the development process.</li> </ul>
<b>Extended Deadline (if needed)</b>	17 November 2025	Optional final submission	Only if there are significant delays or issues (not recommended by GSoC).

## 9. Management of Coding Project

In order to ensure the smooth and efficient development of the Ecotourism data package, I will follow a structured coding management plan. This includes thorough testing strategies, regular updates through a dedicated blog, ongoing contributions even after GSoC, and clear communication with mentors.

### 9.1. Testing Strategy

To maintain the reliability and accuracy of the codebase, I will implement a comprehensive testing strategy, including:

- **Unit Tests:** I will write unit tests using the `testthat` package to verify the correctness of individual functions. Each function will have corresponding test cases to validate its output against expected results.
- **Integration Tests:** To ensure smooth interaction between different modules, I will implement integration tests. This will verify that spatial and temporal matching functions produce consistent and accurate results.
- **Continuous Integration (CI):** I will use GitHub Actions to automate testing on every push, ensuring code quality and early detection of issues.
- **Performance Tests:** I will benchmark the package to ensure it handles large datasets efficiently and remains performant under varying conditions.

### 9.2. Weekly Blog Updates

Throughout the GSoC period, I will maintain a dedicated blog where I will post weekly updates, including:

- Detailed descriptions of my progress, technical challenges faced, and solutions implemented.
- Code snippets and examples to demonstrate the functionality of the package.
- Visualizations and sample outputs showcasing the evolving features of the project.

The blog will also serve as the final project report, making it easier for the mentors to track my progress.



---

### 9.3. Post-GSoC Contributions

My contributions to this project will not end with GSoC. After the official coding period, I plan to:

- Continue enhancing the package by adding new features, improving performance, and fixing any reported bugs.
- Contribute to the R community by collaborating on future improvements and helping maintain the package.
- Engage with the R community through forums and GitHub discussions to support users of the package.

This ongoing involvement will ensure the long-term sustainability and growth of the project.

### 9.4. Communication and Collaboration

To keep the mentors informed about my progress and receive timely feedback, I will:

- Hold weekly Zoom or Google Meet meetings with my mentors to discuss progress, challenges, and next steps.
- Use email and real-time messaging platforms for regular communication and status updates.
- Share milestones and deliverables through GitHub and document them on the blog.

## 10. Test Submissions

As part of the GSoC application process, I completed three test submissions (Easy, Medium, and Hard), which involved geospatial data visualization, density estimation, and geocoding of SA2 regions.

### 10.1. Easy Test:

In this task, I used the `galah` R package to fetch occurrence records from the Atlas of Living Australia. I created a visualization by plotting the occurrences on a map of Australia using:

- `ggplot2` for creating the map.
- `ggspatial` to add spatial context and scale bars.

The final plot displayed the occurrence data accurately, demonstrating the ability to visualize geospatial data effectively.

### 10.2. Medium Test:

This task involved retrieving platypus occurrence data in Victoria, Australia, and associating it with weather data from the nearest weather station. The steps included:



- 
- **Density Estimation:** Applied Kernel Density Estimation (KDE) to identify the most densely populated platypus sighting area, highlighted in purple on the map.
  - **Weather Data Retrieval:** Retrieved daily temperature and precipitation data for 2024 from the nearest weather station using the `GSODR` package.
  - **Visualization:** The final map displayed:
    - Blue Dots: Individual platypus sightings.
    - Purple Region: KDE-based densest area.
    - Green Dot: Nearest weather station.

This task demonstrated my ability to integrate spatial analysis with real-world environmental data.

### 10.3. Hard Test:

In the hard test, I performed geocoding on Statistical Area Level 2 (SA2) regions by calculating their centroids using polygon coordinates from an `ESRI` shapefile. The steps involved:

- **Dataset Cleaning:** Downloaded the domestic trips dataset and removed unnecessary metadata for consistency.
- **Polygon Extraction:** Used `GeoPandas` in Python to extract the centroid coordinates of SA2 regions.
- **Merging Datasets:** Combined the centroid coordinates with the domestic trips dataset to create a spatially-enriched dataset.

The final dataset contained:

- Domestic trip statistics.
- Corresponding SA2 region centroid coordinates.