GeoGrid-Bench: Can Foundation Models Understand Multimodal Gridded Geo-Spatial Data?

Anonymous Authors[1]

Abstract

We present GeoGrid-Bench, a benchmark designed to evaluate the ability of foundation models to understand geo-spatial data in the grid structure. Geo-spatial datasets pose distinct challenges due to their dense numerical values, strong spatial and temporal dependencies, and unique multimodal representations including tabular data, heatmaps, and geographic visualizations. To assess how foundation models can support scientific research in this domain, GeoGrid-Bench features large-scale, real-world data covering 16 climate variables across 150 locations and extended time frames. The benchmark includes approximately 3,200 question-answer pairs, systematically generated from 8 domain expert-curated templates to reflect practical tasks encountered by human scientists. These range from basic queries at a single location and time to complex spatiotemporal comparisons across regions and periods. Our evaluation reveals that vision-language models perform best overall, and we provide a fine-grained analysis of the strengths and limitations of different foundation models in different geo-spatial tasks. This benchmark offers clearer insights into how foundation models can be effectively applied to geo-spatial data analysis and used to support scientific research.[1]

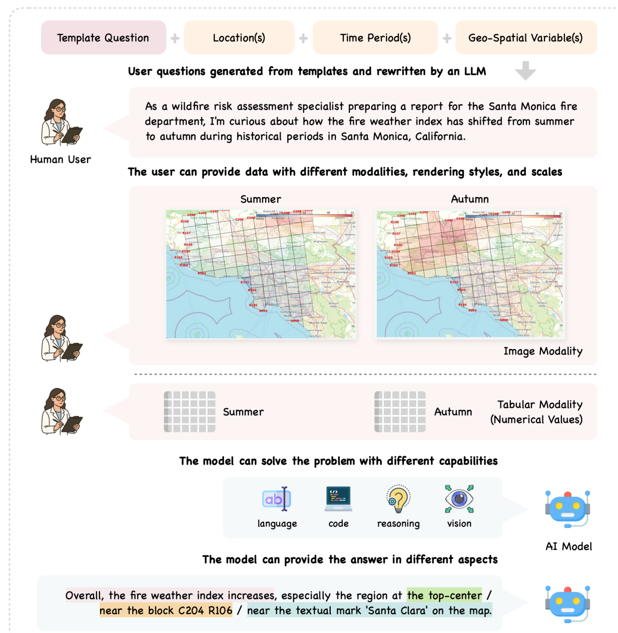


Figure 1. **Overview of GeoGrid-Bench.** The benchmark features questions generated from templates that vary by location, time period, and climate variable, then rewritten with natural language context. Each question is paired with multimodal input—either heatmaps as images or tabular grids of numerical values. We evaluate models on their ability to solve the queries through different modalities—natural language, code, or vision. Ground-truth answers capture find-grained aspects like overall trends, spatial references (from top-left to lower-right), coordinate references (row and column indices), and label references (textual marks on the maps), whenever available.

1. Introduction

Geo-spatial data pose distinct challenges for foundation models due to their inherent spatio-temporal dependencies and exceptionally high data density. Unlike typical tabular records for knowledge retrieval (Zhang et al., 2023a; Pasupat & Liang, 2015; Zhang et al., 2025) or natural images, climate data exists in structured, gridded formats with complex, interconnected numerical values often represented through modalities such as tables, heatmaps, or geographic images spanning across space and time. These data are typically organized in highly structured, gridded formats that encode interconnected numerical values across spatial and temporal dimensions. Each data point is not an isolated unit but part of a dense, multi-dimensional array that reflects physical processes, environmental interactions, or geographical phenomena evolving over time. Meanwhile, models

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]All code and data will be made publicly available upon acceptance.

can also easily get lost in the context (Liu et al., 2023) with overwhelming volumes of values per sample.

Informed decision-making in fields such as disaster response, climate science, and urban development depends on the ability to detect and interpret patterns across regions and over time. However, there remains a lack of benchmarks that directly address the unique challenges posed by geo-spatial gridded data. Most existing efforts docus on object detection, semantic segmentation, object counting, captioning, or scene understanding of Earth observation images (Lacoste et al., 2023; Danish et al., 2024; Zhang & Wang, 2024; Zheng et al., 2023; Wang et al., 2024; Muhtar et al., 2024; Bazi et al., 2024; Kuckreja et al., 2024), function calls to the Geographic Information System (GIS) or SQL queries for data retrieval (Krechetova & Kochedykov, 2025; Jiang & Yang, 2024; Ning et al., 2025; Mooney et al., 2023; Zhang et al., 2023b), or simplified query setups that overlook the spatial-temporal complexities in practical geo-spatial analysis (Bhandari et al., 2023).

To understand how foundation models can assist geo-spatial data analysis, we introduce GeoGrid-Bench, a benchmark explicitly designed to evaluate model performance on multimodal, real-world geo-spatial data. We adopt domain expert-curated query templates to reflect realistic questions that practitioners would encounter in geo-spatial analysis—providing data in both tabular and image formats. These tasks range from simple queries about a fixed location and time to more complex analyses involving multiple locations and temporal comparisons. For each template, we develop oracle code that is applied uniformly to all query instances, enabling scalable and consistent generation of question-answer pairs. Our contributions can be summarized as follows:

- **Large-scale, real-world data:** A domain-centric benchmark built on large-scale, real-world climate projection data, presented in multimodal formats commonly used by actual practitioners, including structured numerical tables and geographic visualizations.
- **Scalable query generation:** A systematic user query generation pipeline based on domain expert-designed templates, reflecting diverse and realistic scientific challenges.
- **Comprehensive evaluation:** Evaluation of foundation models with language, coding, multimodal, and reasoning capabilities across find-grained answer aspects and data modalities to diagnose their strengths and weaknesses in geo-spatial analysis tasks.

Through comprehensive evaluations, we find that visualizing dense, gridded geo-spatial data as heatmaps is the most accessible format for existing foundation models to interpret. In contrast, models struggle to generate flawless code for completing these tasks. Across all model types, identifying broad trends proves easier than making fine-grained regional distinctions, and models exhibit varying strengths and weaknesses depending on the task. With GeoGrid-Bench, we aim to shed light on the strengths and limitations of current foundation models when applied to multimodal geo-spatial data, a core yet underexplored format in climate science. Our goal is to support and advance the development of practical AI-assisted tools that can aid scientific research and decision-making.

2. GeoGrid-Bench: Overview of Data Features and Tasks

GeoGrid-Bench aims to reflect the real-world challenges that scientists face when analyzing geo-spatial data at scale. To achieve this, it features *large-scale, real-world* geo-spatial data sourced and sampled from ClimRR (Argonne National Laboratory, 2023), capturing the complexity of environmental conditions across North America. An overview of user-model interaction is shown in Figure 1.

GeoGrid-Bench is built to capture the unique grid structure. Climate projection data are typically organized across spatial grids and time sequences, resulting in dense, high-dimensional arrays. The data is inherently interconnected, with each point influenced by its geographic neighbors and historical context. This structure poses unique challenges: models must capture spatio-temporal dependencies and handle variability across scales to derive meaningful insights.

Geo-spatial data is also inherently multimodal, presented as tabular data, heatmaps, or geographic visualizations, with each format sharing alignment across a spatial grid structure. Each grid cell encodes a rich array of numerical data that captures localized atmospheric behavior and climate dynamics over time. This multimodal grid structure makes our GeoGrid-Bench an ideal testbed for foundation models designed to reason across space, time, and modality. To perform well, foundation models must integrate spatial context from neighboring cells, understand temporal trends across multi-year projections, and interpret information presented in diverse formats and patterns.

To capture the wide range of questions concerning practitioners at the forefront of geo-spatial analysis, we surveyed 13 domain experts in natural hazard risk domains, resulting in 8 template questions based on their input and around 3200 query instances in GeoGrid-Bench. Each template includes placeholders based at one or two geographic locations, time frames, and climate variables, requiring one to eight data frames. This design allows us to generate a scalable set of scientifically concrete queries that reflect analytical goals. Specifically, GeoGrid-Bench evaluates the following capabilities of foundation models: 1. Identifying regions with the most significant patterns. 2. Comparing data across different locations and times. 3. Analyzing temporal trends and seasonal variations. 4. Interpreting data in multimodal formats.

3. Constructing GeoGrid-Bench At Scale

GeoGrid-Bench features diverse real-world geo-spatial data. Now we discuss our sample curation process, and a visual illustration is included in Figure 4 in the appendix. Each data sample is formed by extracting a specific climate-location-time slice from the ClimRR (Argonne National Laboratory, 2023) dataset. We sample from the 16 climate variables listed in the appendix. For each climate variable, we select around 50 locations where this climate variable is the most prominent, resulting in a total of 150 distinct locations across all climate variables, a subset of ClimRR. For example, the benchmark includes more regions in Southern California for wildfire risk, while precipitation-related examples are more concentrated in the Pacific Northwest to reflect region-specific climate concerns.

We render each data sample in either a **tabular** or **image** format, both structured over a spatial grid. For a given location and its longitude and latitude, we retrieve all grid cells within a square region with edge size 84 to 144 km around it, resulting in approximately 50 to 150 entries in the 12-by-12 km grid. In the **tabular** modality, we prepare each table with numerical values, a caption, and row and column indices as textual strings. In the **image** modality, we prepare three types of visualization with increasing information densities: (1) A standalone heatmap, (2) A heatmap with overlaid numerical annotations at each grid cell, and (3) A heatmap overlaid on an actual geographic base map. Specifically, we render the tabular data as a heatmap with color gradients. This heatmap is optionally added with numerical annotation of the value on each cell, or overlaid on a geographic base map (OpenStreetMap contributors, 2024) using Folium (Folium, 2023). To maintain consistency with the tabular format, we also render row and column indices around the heatmap. This visualization offers a richer representation to mirror common practices in real-world analysis. To isolate the challenge of data retrieval, GeoGrid-Bench provides the foundation model during evaluation with all necessary data frames in either tabular or image formats, focusing solely on whether the model can solve the problem given the relevant information.

GeoGrid-Bench builds on expert-curated templates for scalable query generation. Based on in-depth discussions about the analytical tasks our domain experts perform, we develop eight representative question templates, which are included in Table 1 in the appendix. Each template takes as input one or two climate variables, locations, and time frames and outputs a filled-in user query in our benchmark, and may require between one and eight data frames to answer. This structured approach enables the automatic generation of a wide variety of concrete, data-driven queries tailored to real-world analytical needs.

For every template, we manually craft oracle code that deterministically solves the question and prepares ground-truth answers in desired formats. *Crucially, the same oracle applies uniformly to every query generated from a given template, enabling the scalability of the benchmark. As a result, once a template and its oracle are validated, we ensure the quality of every generated instance.*

Each question is a multiple-choice with four options, all generated by the oracle code rather than a language model. Recognizing that a foundation model may excel at different aspects in answering a geo-spatial query, each query probes a different aspect in giving the answer, as shown in Figure 1. Specifically, answer options target the following aspects: (1) Overall patterns (e.g., the wildfire risk overall increases). (2) Spatial references (e.g., the highest wildfire risk occurs around the top-left region). (3) Coordinate references (e.g., the highest wildfire risk occurs around Column 204 Row 106). (4) Label references (e.g., the highest wildfire risk occurs near the textual label ”Santa Clara” on the map), which is only available for the image type ”heatmap overlaid on an actual geographic base map”. In addition, to explore which data modalities most effectively support geo-spatial analysis, we evaluate models across three input settings: **language-only**, **language and code**, and **language and vision**. Detailed prompting and result parsing strategies for each setting are provided in Appendix C.

4. Experiment

Experimental Setup. We benchmark a range of state-of-the-art closed-source and open-source models on GeoGrid-Bench. Our evaluation covers 5 models from OpenAI, including o4-mini, GPT-4.1, GPT-4.1-mini, GPT-4o, and GPT-4o-mini (OpenAI, 2024; 2025; Hurst et al., 2024), and 6 open-source models including Llama-4-Maverick, Llama-4-Scout, Llama-3.2-11B-Vision, Llama-3.2-3B, Llama-3.1-8B (Grattafiori et al., 2024; AI, 2024), and Qwen-2.5-VL-7B (Bai et al., 2025). OpenAI models are accessed via API calls, and Llama-4 models are accessed through the Lambda Inference API. Inferences for other open-source models run locally on four NVIDIA A100-SXM4 GPUs with 40GB of VRAM. For all models, we set `max_new_tokens` as 1024 with default temperature and sampling strategies.

4.1. Evaluation Results and Findings

Vision-language models achieve the strongest performance in geo-spatial tasks. Among the models we evalu-

model_name	data_modality	overall_accuracy	trend
o4-mini	language and vision	0.644	0.686
GPT-4.1	language and vision	0.578	0.668
GPT-4.1-mini	language and vision	0.568	0.655
o4-mini	language-only	0.534	0.474
GPT-4o	language and vision	0.518	0.599
GPT-4.1	language-only	0.512	0.505
model_name	**data_modality**	**overall_accuracy**	**trend**
Llama-4-Maverick	language and vision	0.580	0.696
Llama-4-Scout	language and vision	0.508	0.607
Llama-4-Maverick	language-only	0.486	0.517
Llama-4-Scout	language-only	0.457	0.478
Qwen2.5-VL-7B	language and vision	0.413	0.445
Llama-4-Maverick	language and code	0.337	0.282

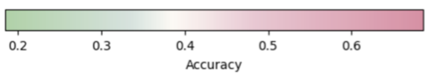


Figure 2. Selected evaluation results for OpenAI and open-source models across different data modalities (columns correspond to the fine-grained answer aspects defined in Section 3). This table shows only a subset of the full results; the complete evaluation tables can be found in the Appendix.

ate, o4-mini achieves overall the highest performance, while Llama-4-Maverick leads among open-source models, as shown in Figure 5 in the Appendix. Overall, models that receive input in the vision modality consistently outperform those using language-only input. This suggests that converting geo-spatial gridded data into heatmap visualizations—rather than presenting models directly with large volumes of raw numerical values in tabular forms—enables foundation models to more effectively interpret such data with complex spatial-temporal patterns.

Inferior performance in code highlights the need for more agentic models in geo-spatial tasks. Contrary to our expectations, foundation models leveraging programming code do not outperform their language-only counterparts on our task. Upon closer inspection, much of the generated code is not directly executable in a single pass. For instance, models produce incomplete scripts or bugs, omit expected outputs, fail to parse data, or struggle with planning over geo-spatial data—ultimately requiring human intervention across multiple iterations. This limitation aligns with how we construct the oracle code in the benchmark. This issue is more severe in open-source models like Llama, which tend to produce fewer executable code. We, therefore, emphasize the need for stronger *agentic* behaviors (Plaat et al., 2025; Kapoor et al., 2024; Ng, 2024) in foundation models, where we define ”agentic” as the ability to autonomously generate fully executable code for human end-users in a single interaction, particularly when the end-users are domain scientists rather than programmers.

Fine-grained geo-spatial tasks reveals different strength-weakness tradeoffs. Commercial and open-source models exhibit different strengths and weaknesses in fine-grained geo-spatial tasks, as shown in Figure 5 in the Appendix. Specifically, open-source models generally struggle more than commercial ones in identifying regions with the most significant patterns. However, both types of models perform well when comparing trends between two locations or analyzing seasonal variations at a single location. In contrast, they show weaker performance when comparing seasonal variations across multiple locations or comparing data across different locations and times.

Models perform better at identifying overall trends than fine-grained region detections. As mentioned in Figure 1, target answers captures fine-grained aspects in answering these geo-spatial queries. Evaluation results in Figure 6 (a) and (b) in the Appendix show that models perform best on the ”trend” column, while accuracy drops for spatial, coordinate, or label references—highlighting a need for improvement in fine-grained regional understanding.

Heatmaps with numerical annotations enhance performance, whereas map-overlaid heatmaps pose greater challenges for vision-language models. Figure 6 (c) in the Appendix compares model performance across three input image formats. Adding numerical annotations to heatmaps improves model accuracy compared to using color gradients alone. In contrast, the most realistic format, where heatmaps are overlaid on geographic base maps, poses the challenge for all models, as the added visual complexity hinders spatial pattern recognition.

5. Conclusion

We introduced GeoGrid-Bench, a comprehensive benchmark designed to evaluate the capability of foundation models to understand multimodal gridded geo-spatial data. GeoGrid-Bench features structured, dense numerical data using real-world gridded datasets and expert-curated templates to evaluate scientifically relevant geo-spatial tasks. This integrated design enables robust and scalable assessment of foundation models across vision, language, and code modalities. Our evaluation reveals that while vision-language models excel at interpreting spatial patterns from heatmaps, they still struggle with fine-grained regional understanding and label-based reasoning. Meanwhile, language and code models show limited success in generating executable analysis scripts without human intervention, highlighting the need for stronger agentic behavior. These findings point to several critical areas where model capabilities must improve to meet the practical needs of geo-spatial scientific analysis.

Impact Statement

Overall, this work can inform the development of more capable models to process and understand the dense numerical data, spatiotemporal dependencies, and multimodal representations of geo-spatial data, supporting the advancement of foundation models for informed decision-making and resilience building across a wide range of real-world challenges.

We acknowledge that this dataset is limited to the United States due to data availability. Additionally, our benchmark focuses on geo-spatial data in gridded formats, intentionally excluding other common data types such as Earth observation and remote sensing imagery, which have already been extensively studied in prior work. However, the underlying framework are designed to be generalizable and can be readily applied to similar gridded geo-spatial datasets from other regions. Building on this foundation, future work will focus on expanding GeoGrid-Bench beyond the United States and incorporating richer data modalities such as satellite imagery, elevation maps, and land use data to enable broader and more diverse analytical capabilities.

References

AI, M. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, 2024. URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/. Accessed: 2025-04-27.

Argonne National Laboratory. Climrr: Climate risk and resilience portal. https://climrr.anl.gov, 2023. Accessed: 2025-04-15.

Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Bazi, Y., Bashmal, L., Al Rahhal, M. M., Ricci, R., and Melgani, F. Rs-llava: A large vision-language model for joint captioning and question answering in remote sensing imagery. *Remote Sensing*, 16(9):1477, 2024.

Bhandari, P., Anastasopoulos, A., and Pfoser, D. Are large language models geospatially knowledgeable? In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pp. 1–4, 2023.

Danish, M. S., Munir, M. A., Shah, S. R. A., Kuckreja, K., Khan, F. S., Fraccaro, P., Lacoste, A., and Khan, S. Geobench-vlm: Benchmarking vision-language models for geospatial tasks. *arXiv preprint arXiv:2411.19325*, 2024.

Folium. Folium: Python data. leaflet.js maps. https://python-visualization.github.io/folium/latest/, 2023. URL https://python-visualization.github.io/folium/latest/. Version 0.14.0.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Jiang, Y. and Yang, C. Is chatgpt a good geospatial data analyst? exploring the integration of natural language into structured query language within a spatial database. *ISPRS International Journal of Geo-Information*, 13(1): 26, 2024.

Kapoor, S., Stroebl, B., Siegel, Z. S., Nadgir, N., and Narayanan, A. Ai agents that matter. *arXiv preprint arXiv:2407.01502*, 2024.

Krechetova, V. and Kochedykov, D. Geobenchx: Benchmarking llms for multistep geospatial tasks. *arXiv preprint arXiv:2503.18129*, 2025.

Kuckreja, K., Danish, M. S., Naseer, M., Das, A., Khan, S., and Khan, F. S. Geochat: Grounded large vision-language model for remote sensing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27831–27840, 2024.

Lacoste, A., Lehmann, N., Rodriguez, P., Sherwin, E., Kerner, H., Lütjens, B., Irvin, J., Dao, D., Alemohammad, H., Drouin, A., et al. Geo-bench: Toward foundation models for earth monitoring. *Advances in Neural Information Processing Systems*, 36:51080–51093, 2023.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.

Mooney, P., Cui, W., Guan, B., and Juhász, L. Towards understanding the geospatial skills of chatgpt: Taking a geographic information systems (gis) exam. In *Proceedings of the 6th ACM SIGSPAtIAL international workshop on AI for geographic knowledge discovery*, pp. 85–94, 2023.

Muhtar, D., Li, Z., Gu, F., Zhang, X., and Xiao, P. Lhrs-bot: Empowering remote sensing with vgi-enhanced large multimodal language model. In *European Conference on Computer Vision*, pp. 440–457. Springer, 2024.

Ng, A. Welcoming diverse approaches keeps machine learning strong. June 2024. URL https://www.deeplearning.ai/the-batch/welcoming-diverse-approaches-keeps-machine-learning-strong/.

Ning, H., Li, Z., Akinboyewa, T., and Lessani, M. N. An autonomous gis agent framework for geospatial data retrieval. *International Journal of Digital Earth*, 18(1): 2458688, 2025.

OpenAI. Openai o3 and o4-mini system card, 2024. URL https://openai.com/index/o3-o4-mini-system-card/. Accessed: 2025-04-18.

OpenAI. Introducing gpt-4.1 in the api, 2025. URL https://openai.com/index/gpt-4-1/. Accessed: 2025-05-12.

OpenStreetMap contributors. Openstreetmap, 2024. URL https://www.openstreetmap.org/. Accessed: 2025-05-12.

Pasupat, P. and Liang, P. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*, 2015.

Plaat, A., van Duijn, M., van Stein, N., Preuss, M., van der Putten, P., and Batenburg, K. J. Agentic large language models, a survey. *arXiv preprint arXiv:2503.23037*, 2025.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Wang, J., Zheng, Z., Chen, Z., Ma, A., and Zhong, Y. Earthvqa: Towards queryable earth via relational reasoning-based remote sensing visual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5481–5489, 2024.

Zhang, C. and Wang, S. Good at captioning bad at counting: Benchmarking gpt-4v on earth observation data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7839–7849, 2024.

Zhang, T., Yue, X., Li, Y., and Sun, H. Tablellama: Towards open large generalist models for tables. *arXiv preprint arXiv:2311.09206*, 2023a.

Zhang, X., Wang, D., Dou, L., Zhu, Q., and Che, W. A survey of table reasoning with large language models. *Frontiers of Computer Science*, 19(9):199348, 2025.

Zhang, Y., Wei, C., Wu, S., He, Z., and Yu, W. Geogpt: Understanding and processing geospatial tasks through an autonomous gpt. *arXiv preprint arXiv:2307.07930*, 2023b.

Zheng, H., Zhang, C., Guan, K., Deng, Y., Wang, S., Rhoads, B. L., Margenot, A. J., Zhou, S., and Wang, S. Segment any stream: Scalable water extent detection with the segment anything model. In *NeurIPS 2023 Computational Sustainability: Promises and Pitfalls from Theory to Deployment*, 2023.

A. Data Curation

We illustrate our sample curation process in Figure 4. Our template questions are included in Table 1. The full list of Climate Variables in GeoGrid-Benchis included below.

Full List of Climate Variables in GeoGrid-Bench

Maximum Annual Temperature, Minimum Annual Temperature, Consecutive Days with No Precipitation, Cooling Degree Days, Fire Weather Index, Maximum Daily Heat Index, Maximum Seasonal Heat Index, Number of Days with Daily Heat Index ¿ 95°F/105°F/115°F/125°F, Heating Degree, Annual Total Precipitation, Maximum Seasonal Temperature, Minimum Seasonal Temperature, Wind Speed.

Templates that require one data frame
1. Which region in the {location1} experienced the largest increase in {variable1} during {time_frame1}?
Templates that require two data frames
2. How has {variable1} changed between {time_frame1} and {time_frame2} in the {location1}?
3. What is the correlation between {variable1} and {variable2} in the {location1} during {time_frame1}?
4. How does {variable1} compare between {location1} and {location2} during {time_frame1}?
Templates that require four data frames
5. What is the *seasonal* variation of {climate_variable1} in {location1} during {time_frame1}?
6. Which *season* in {time_frame1} saw the highest levels of {variable1} in {location1}?
7. Which of {location1} or {location2} experienced a greater change in {variable1} throughout {time_frame1} and {time_frame2}?
Templates that require eight data frames
8. How does the *seasonal* variation of {variable1} in {location1} compare to that in {location2} for {time_frame1}?

Table 1. **Template questions in GeoGrid-Bench.** We develop those questions with domain experts. Each question includes placeholders for one or two locations, time frames, and geo-spatial variables. This design enables scalable question construction while capturing varying levels of complexity based on the number of data frames involved.

B. Evaluation Results

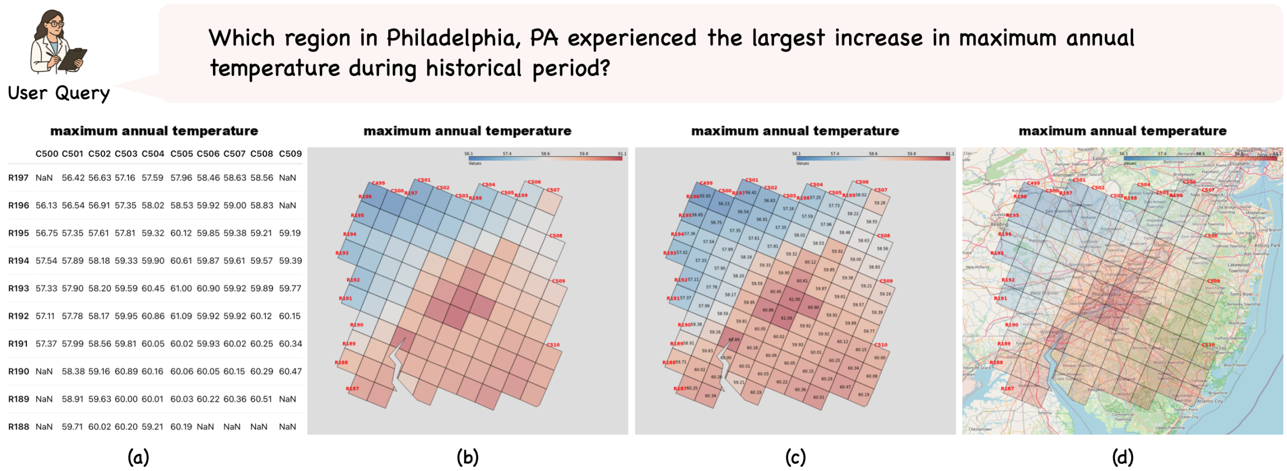
The evalution result tables are shown in Figure 5 and 6.

C. Inference Prompts and Result Parsing

To evaluate models across different modalities, we design prompts for three settings: language-only, language and code, and language and vision. Each prompt is designed to be simple yet encourage model response with desired style and consistent answer formatting.

- Language-only: models receive data in tabular format with instructions *"Think step by step before making a decision. Then, explicitly state your final choice after the special phrase "####Final Answer" followed by (a), (b), (c), or (d). Please don't use programming code."*.
- Language and programming code: models receive data in tabular format with instructions *"Please write Python code to answer the question and show the complete script. You must include a print statement at the end of the code that outputs the final answer using the special phrase "####Final Answer' followed by (a), (b), (c), or (d)."*
- Language and vision: models receive climate data in one of the three image formats with instructions *"Analyze this image and answer the question. Think step by step before making a decision. Then, explicitly state your final choice after the special phrase "####Final Answer" followed by (a), (b), (c), or (d)."*.

In each mode, we provide the model with the user query, the relevant data (in either tabular or image format), all four multiple-choice options, and system instructions as inputs. We extract the model's final answer following the special tokens



	C500	C501	C502	C503	C504	C505	C506	C507	C508	C509
R197	NaN	56.42	56.63	57.16	57.59	57.96	58.46	58.63	58.56	NaN
R196	56.13	56.54	56.91	57.35	58.02	58.53	59.92	59.00	58.83	NaN
R195	56.75	57.35	57.61	57.81	59.32	60.12	59.85	59.38	59.21	59.19
R194	57.54	57.89	58.18	59.33	59.90	60.61	59.87	59.61	59.57	59.39
R193	57.33	57.90	58.20	59.59	60.45	61.00	60.90	59.92	59.89	59.77
R192	57.11	57.78	58.17	59.95	60.86	61.09	59.92	59.92	60.12	60.15
R191	57.37	57.99	58.56	59.81	60.05	60.02	59.93	60.02	60.25	60.34
R190	NaN	58.38	59.16	60.89	60.16	60.06	60.05	60.15	60.29	60.47
R189	NaN	58.91	59.63	60.00	60.01	60.03	60.22	60.36	60.51	NaN
R188	NaN	59.71	60.02	60.20	59.21	60.19	NaN	NaN	NaN	NaN

Figure 3. We prepare every data sample in one of the four formats: (a) 2D table as a textual string. (b) standalone heatmap; (c) heatmap with overlaid numerical annotations at each grid cell; (d) heatmap overlaid on an actual geographic base map. These formats reflect real-world climate data practices and differ markedly from typical natural images seen by foundation models. More in Appendix D.

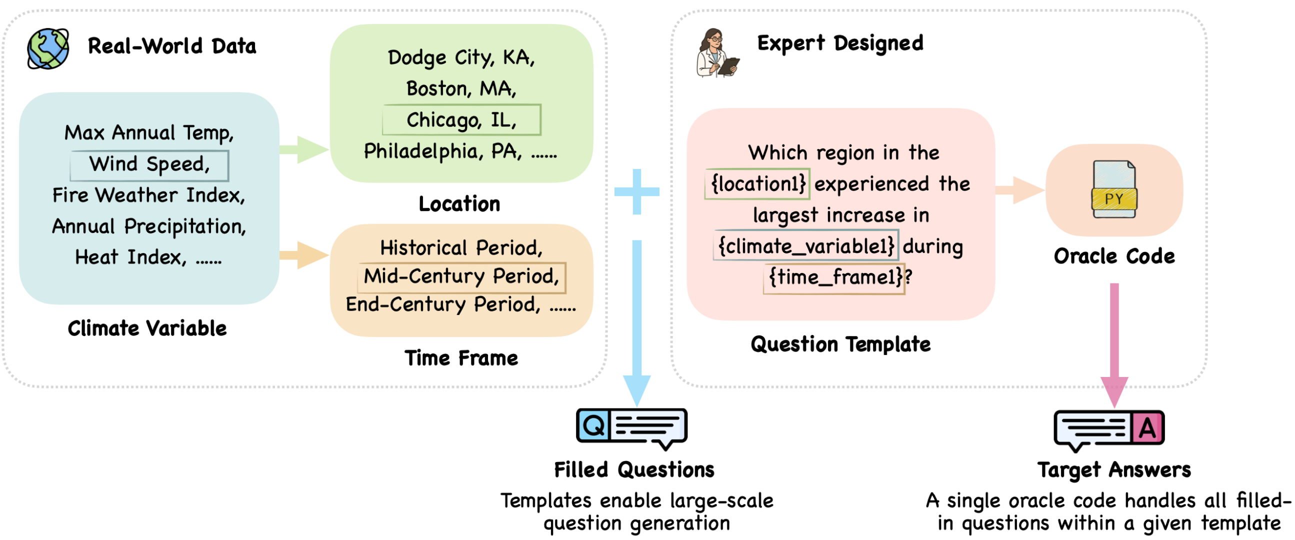


Figure 4. **Overview of the example curation process.** Each example in GeoGrid-Bench is constructed by combining a query template with sampled climate variables, locations, and time frames from real-world climate data. Each template is paired with a corresponding oracle code that deterministically generates target answers for all filled-in question instances under that template.

"####Final Answer" to facilitate answer parsing. If the model fails to provide an explicit option (a), (b), (c), or (d), we use a sentence embedding model (Reimers & Gurevych, 2019) to identify the most similar option based on the model's response. When the model outputs Python code, we execute the code in a shell environment to extract the final answers.

D. Examples of Data Visualizations for All Query Templates

model_name	data_modality	overall_accuracy	Which of {location1} or {location2} experienced a greater change in {climate_variable1} throughout {time_frame1} and {time_frame2}?	What is the seasonal variation of {climate_variable1} in {location1} during {time_frame1}?	Which region in the {location1} experienced the largest increase in {climate_variable1} during {time_frame1}?	Which season in {time_frame1} saw the highest levels of {climate_variable1} in {location1}?	How does {climate_variable1} compare between {location1} and {location2} during {time_frame1}?	What is the correlation between {climate_variable1} and {climate_variable2} in the {location1} during {time_frame1}?	How has {climate_variable1} changed between {time_frame1} and {time_frame2} in the {location1}?	How does the seasonal variation of {climate_variable1} in {location1} compare to that in {location2} for {time_frame1}?
o4-mini	language and vision	0.644	0.667	0.673	0.743	0.813	0.623	0.453	0.453	0.724
GPT-4.1	language and vision	0.578	0.640	0.593	0.660	0.823	0.523	0.313	0.400	0.673
GPT-4.1-mini	language and vision	0.568	0.633	0.517	0.600	0.803	0.487	0.373	0.453	0.680
o4-mini	language-only	0.534	0.790	0.800	0.470	0.210	0.590	0.570	0.510	0.333
GPT-4o	language and vision	0.518	0.613	0.407	0.630	0.773	0.447	0.370	0.380	0.525
GPT-4.1	language-only	0.512	0.690	0.670	0.450	0.530	0.540	0.470	0.450	0.293
GPT-4.1-mini	language-only	0.511	0.640	0.670	0.450	0.500	0.580	0.440	0.470	0.333
GPT-4o-mini	language and vision	0.462	0.573	0.657	0.363	0.700	0.400	0.437	0.373	0.192
o4-mini	language and code	0.453	0.650	0.660	0.420	0.150	0.500	0.470	0.530	0.242
GPT-4o-mini	language-only	0.437	0.630	0.550	0.410	0.400	0.270	0.570	0.380	0.283
GPT-4.1-mini	language and code	0.427	0.470	0.570	0.560	0.200	0.410	0.440	0.420	0.343
GPT-4o	language-only	0.423	0.630	0.420	0.430	0.400	0.370	0.420	0.430	0.283
GPT-4.1	language and code	0.412	0.500	0.580	0.350	0.290	0.430	0.420	0.440	0.283
GPT-4o-mini	language and code	0.369	0.440	0.530	0.270	0.260	0.330	0.400	0.390	0.333
GPT-4o	language and code	0.367	0.470	0.520	0.340	0.250	0.330	0.350	0.310	0.364
Overall	language and vision	0.554	0.625	0.569	0.599	0.783	0.496	0.389	0.412	0.559
Overall	language-only	0.483	0.676	0.622	0.442	0.408	0.470	0.494	0.448	0.305
Overall	language and code	0.406	0.506	0.572	0.388	0.230	0.400	0.416	0.418	0.313
Overall	all	0.481	0.602	0.588	0.476	0.474	0.455	0.433	0.426	0.392

model_name	data_modality	overall_accuracy	What is the seasonal variation of {climate_variable1} in {location1} during {time_frame1}?	Which of {location1} or {location2} experienced a greater change in {climate_variable1} throughout {time_frame1} and {time_frame2}?	Which season in {time_frame1} saw the highest levels of {climate_variable1} in {location1}?	What is the correlation between {climate_variable1} and {climate_variable2} in the {location1} during {time_frame1}?	How does {climate_variable1} compare between {location1} and {location2} during {time_frame1}?	How does the seasonal variation of {climate_variable1} in {location1} compare to that in {location2} for {time_frame1}?	How has {climate_variable1} changed between {time_frame1} and {time_frame2} in the {location1}?	Which region in the {location1} experienced the largest increase in {climate_variable1} during {time_frame1}?
Llama-4-Maverick	language and vision	0.580	0.627	0.667	0.807	0.373	0.467	0.721	0.443	0.537
Llama-4-Scout	language and vision	0.508	0.463	0.607	0.727	0.517	0.393	0.556	0.378	0.423
Llama-4-Maverick	language-only	0.486	0.570	0.620	0.460	0.370	0.540	0.424	0.470	0.430
Llama-4-Scout	language-only	0.457	0.490	0.480	0.530	0.510	0.410	0.364	0.402	0.470
Qwen2.5-VL-7B	language and vision	0.413	0.440	0.420	0.507	0.523	0.330	0.380	0.337	0.367
Llama-4-Maverick	language and code	0.337	0.630	0.350	0.100	0.320	0.410	0.192	0.400	0.290
Llama-3.2-3B	language-only	0.312	0.290	0.360	0.410	0.280	0.270	0.293	0.280	0.240
Llama-4-Scout	language and code	0.311	0.470	0.370	0.200	0.310	0.270	0.182	0.347	0.340
Qwen2.5-VL-7B	language-only	0.298	0.370	0.350	0.190	0.420	0.310	0.300	0.260	0.180
Qwen2.5-VL-7B	language and code	0.286	0.310	0.550	0.320	0.180	0.290	0.140	0.300	0.200
Llama-3.2-3B	language and code	0.265	0.620	0.070	0.020	0.240	0.310	0.343	0.230	0.290
Llama-3.1-8B	language and code	0.264	0.670	0.060	0.030	0.260	0.270	0.313	0.240	0.270
Llama-3.2-11B-Vision	language and code	0.261	0.690	0.119	0.034	0.281	0.220	0.306	0.230	0.250
Llama-3.2-11B-Vision	language and vision	0.233	0.258	0.319	0.403	0.281	0.231	0.173	0.277	0.273
Llama-3.2-11B-Vision	language-only	0.204	0.270	0.250	0.310	0.110	0.100	0.160	0.170	0.160
Llama-3.1-8B	language-only	0.173	0.250	0.290	0.330	0.240	0.210	0.172	0.240	0.210
Overall	language and vision	0.433	0.447	0.503	0.611	0.424	0.355	0.457	0.359	0.400
Overall	language-only	0.322	0.373	0.392	0.372	0.322	0.307	0.285	0.304	0.282
Overall	language and code	0.287	0.565	0.253	0.117	0.265	0.295	0.246	0.291	0.273
Overall	all	0.337	0.464	0.368	0.336	0.326	0.314	0.314	0.313	0.308

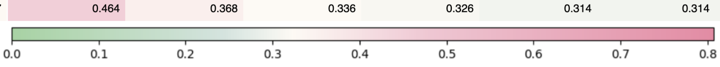


Figure 5. Evaluation results. The top table shows OpenAI models and the bottom table shows open-source models. Each row corresponds to one model with one data modality—language-only, language and code, or language and vision, while each column represents a query template in Table 1.

model_name	data_modality	overall_accuracy	trend	spatial_ref	coordinate_ref	label_ref
o4-mini	language and vision	0.644	0.686	0.565	0.515	0.667
GPT-4.1	language and vision	0.578	0.668	0.451	0.451	0.655
GPT-4.1-mini	language and vision	0.568	0.655	0.475	0.426	0.529
o4-mini	language-only	0.534	0.474	0.426	0.555	nan
GPT-4o	language and vision	0.518	0.599	0.457	0.412	0.517
GPT-4.1	language-only	0.512	0.505	0.361	0.597	nan
GPT-4.1-mini	language-only	0.511	0.482	0.333	0.639	nan
GPT-4o-mini	language and vision	0.462	0.508	0.306	0.300	0.425
o4-mini	language and code	0.453	0.369	0.398	0.546	nan
GPT-4o-mini	language-only	0.437	0.448	0.278	0.437	nan
GPT-4.1-mini	language and code	0.426	0.340	0.417	0.588	nan
GPT-4o	language-only	0.423	0.436	0.185	0.555	nan
GPT-4.1	language and code	0.412	0.371	0.324	0.513	nan
GPT-4o-mini	language and code	0.369	0.366	0.343	0.328	nan
GPT-4o	language and code	0.367	0.356	0.343	0.462	nan

(a)

model_name	data_modality	overall_accuracy	trend	spatial_ref	coordinate_ref	label_ref
Llama-4-Maverick	language and vision	0.580	0.696	0.426	0.375	0.483
Llama-4-Scout	language and vision	0.508	0.607	0.361	0.308	0.414
Llama-4-Maverick	language-only	0.486	0.517	0.287	0.472	nan
Llama-4-Scout	language-only	0.457	0.478	0.269	0.463	nan
Qwen2.5-VL-7B	language and vision	0.413	0.445	0.380	0.342	0.345
Llama-4-Maverick	language and code	0.337	0.282	0.231	0.407	nan
Llama-3.2-3B	language-only	0.312	0.346	0.250	0.296	nan
Llama-4-Scout	language and code	0.311	0.276	0.333	0.343	nan
Qwen2.5-VL-7B	language-only	0.298	0.309	0.148	0.250	nan
Qwen2.5-VL-7B	language and code	0.286	0.317	0.222	0.250	nan
Llama-3.2-3B	language and code	0.265	0.202	0.361	0.343	nan
Llama-3.1-8B	language and code	0.264	0.183	0.361	0.352	nan
Llama-3.2-11B-Vision	language and code	0.261	0.191	0.330	0.330	nan
Llama-3.2-11B-Vision	language and vision	0.233	0.297	0.274	0.267	0.176
Llama-3.2-11B-Vision	language-only	0.204	0.232	0.167	0.130	nan
Llama-3.1-8B	language-only	0.173	0.256	0.250	0.176	nan

(b)

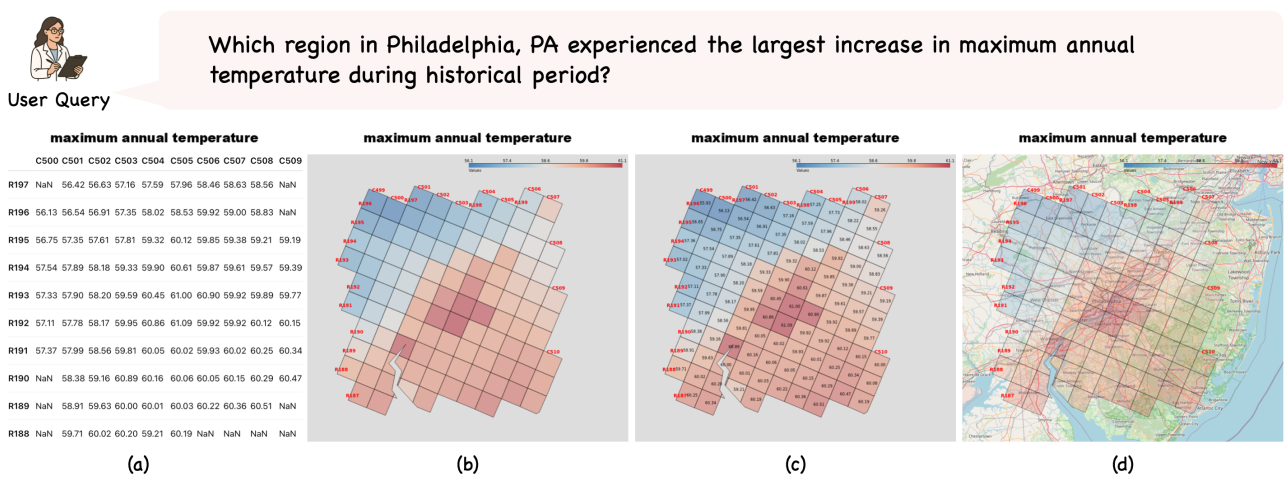
0.2 0.3 0.4 0.5 0.6
Accuracy

model	data_modality	overall_accuracy	heatmap	heatmap_with_annotations	heatmap_overlayed_on_map
o4-mini	language and vision	0.644	0.653	0.700	0.578
Llama-4-Maverick	language and vision	0.580	0.578	0.641	0.521
GPT-4.1	language and vision	0.578	0.573	0.625	0.537
GPT-4.1-mini	language and vision	0.568	0.568	0.620	0.517
GPT-4o	language and vision	0.518	0.524	0.573	0.457
Llama-4-Scout	language and vision	0.507	0.506	0.552	0.464
GPT-4o-mini	language and vision	0.462	0.464	0.501	0.422
Qwen2.5-VL-7B	language and vision	0.413	0.416	0.458	0.365
Llama-3.2-11B-Vision	language and vision	0.277	0.289	0.273	0.268

(c)

0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65
Accuracy

Figure 6. More evaluation results. (a) OpenAI models and (b) open-source models evaluated under different data modalities. Columns represent fine-grained answer aspects defined in Section 3, including trend, spatial references, coordinate references, and label references. There exist NaN values since the label reference is only available for the vision modality. (c) vision-language models, which are evaluated on three visualization types, as mentioned in Section 3 and Figure 3.



	C500	C501	C502	C503	C504	C505	C506	C507	C508	C509
R197	NaN	56.42	56.63	57.16	57.59	57.96	58.46	58.63	58.56	NaN
R196	56.13	56.54	56.91	57.35	58.02	58.53	59.92	59.00	58.83	NaN
R195	56.75	57.35	57.61	57.81	59.32	60.12	59.85	59.38	59.21	59.19
R194	57.54	57.89	58.18	59.33	59.90	60.61	59.87	59.61	59.57	59.39
R193	57.33	57.90	58.20	59.59	60.45	61.00	60.90	59.92	59.89	59.77
R192	57.11	57.78	58.17	59.95	60.86	61.09	59.92	59.92	60.12	60.15
R191	57.37	57.99	58.56	59.81	60.05	60.02	59.93	60.02	60.25	60.34
R190	NaN	58.38	59.16	60.89	60.16	60.06	60.05	60.15	60.29	60.47
R189	NaN	58.91	59.63	60.00	60.01	60.03	60.22	60.36	60.51	NaN
R188	NaN	59.71	60.02	60.20	59.21	60.19	NaN	NaN	NaN	NaN

Figure 7. We prepare every data sample in one of the four formats: (a) 2D table as a textual string. (b) standalone heatmap; (c) heatmap with overlaid numerical annotations at each grid cell; (d) heatmap overlaid on an actual geographic base map. These formats reflect real-world climate data practices and differ markedly from typical natural images seen by foundation models.

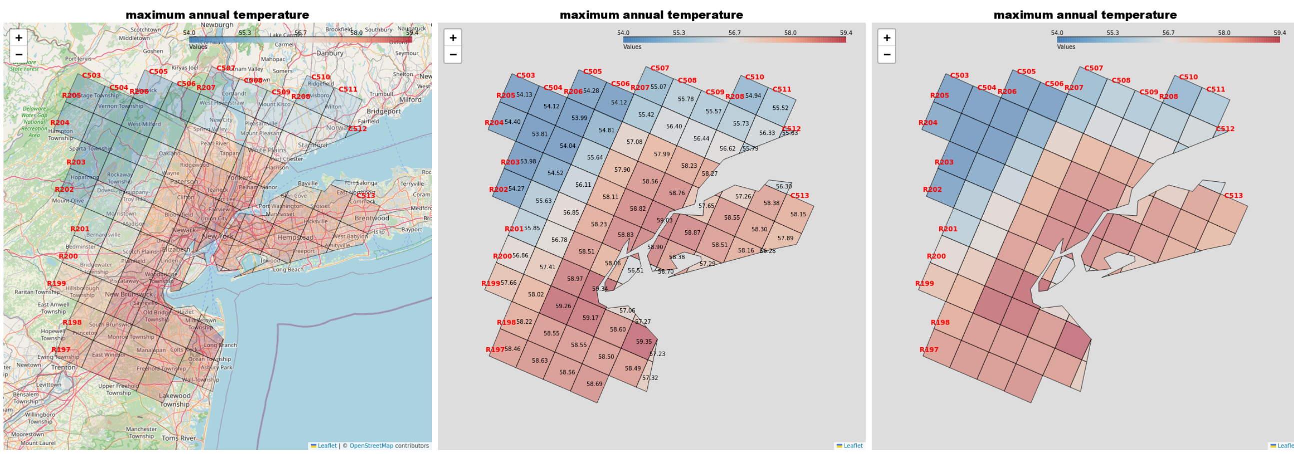


Figure 8. **Template 1:** Which region in {location1} experienced the largest increase in {climate_variable1} during {time_frame1}? This example takes location1 = New York city, NY, climate_variable1 = maximum annual temperate, and time_frame1 = historical period.

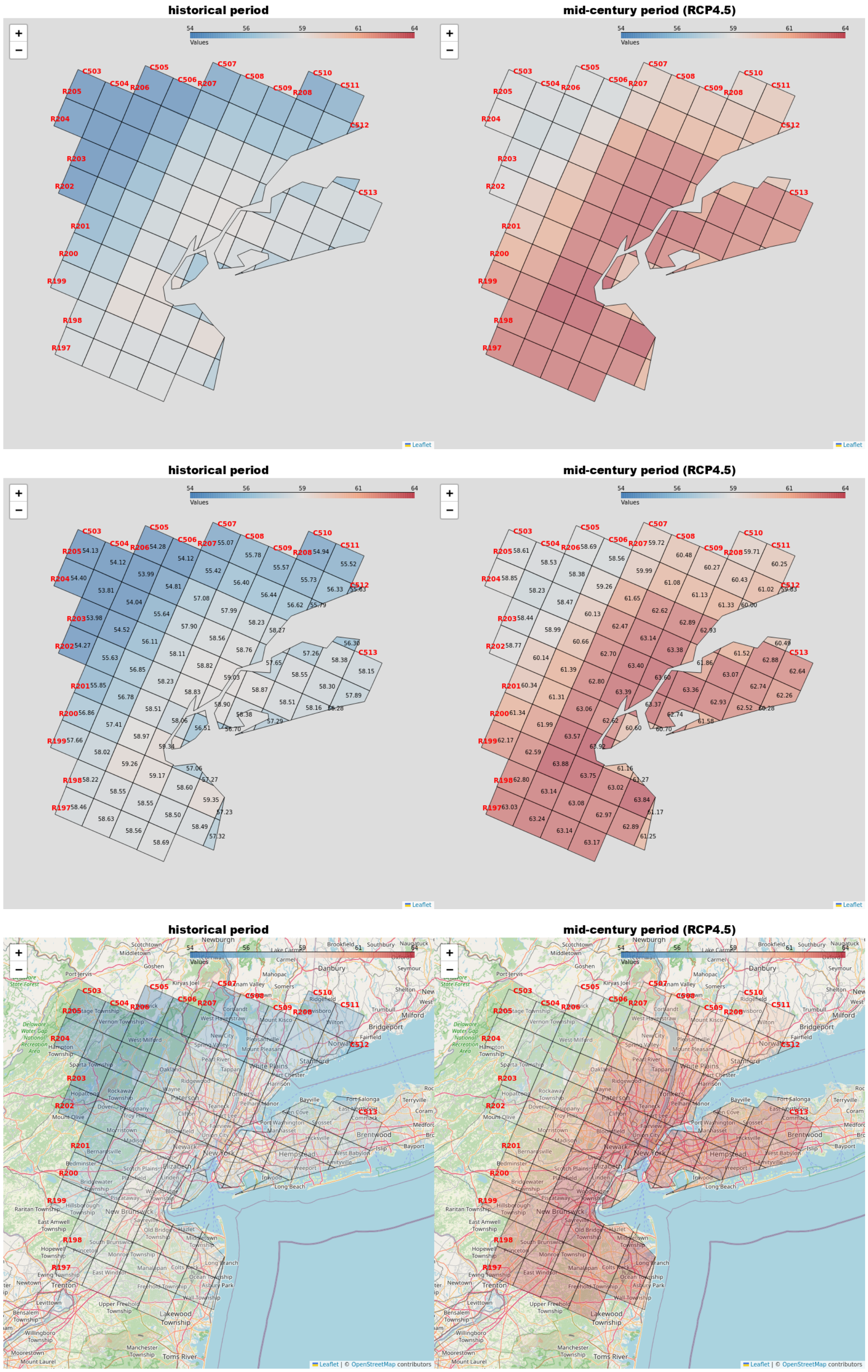


Figure 9. **Template 2:** How has {climate_variable1} changed between {time_frame1} and {time_frame2} in the {location1}? This example takes location1 = New York city, NY, climate_variable1 = maximum annual temperate, time_frame1 = historical period, and time_frame2 = mid-century period (RCP-4.5).

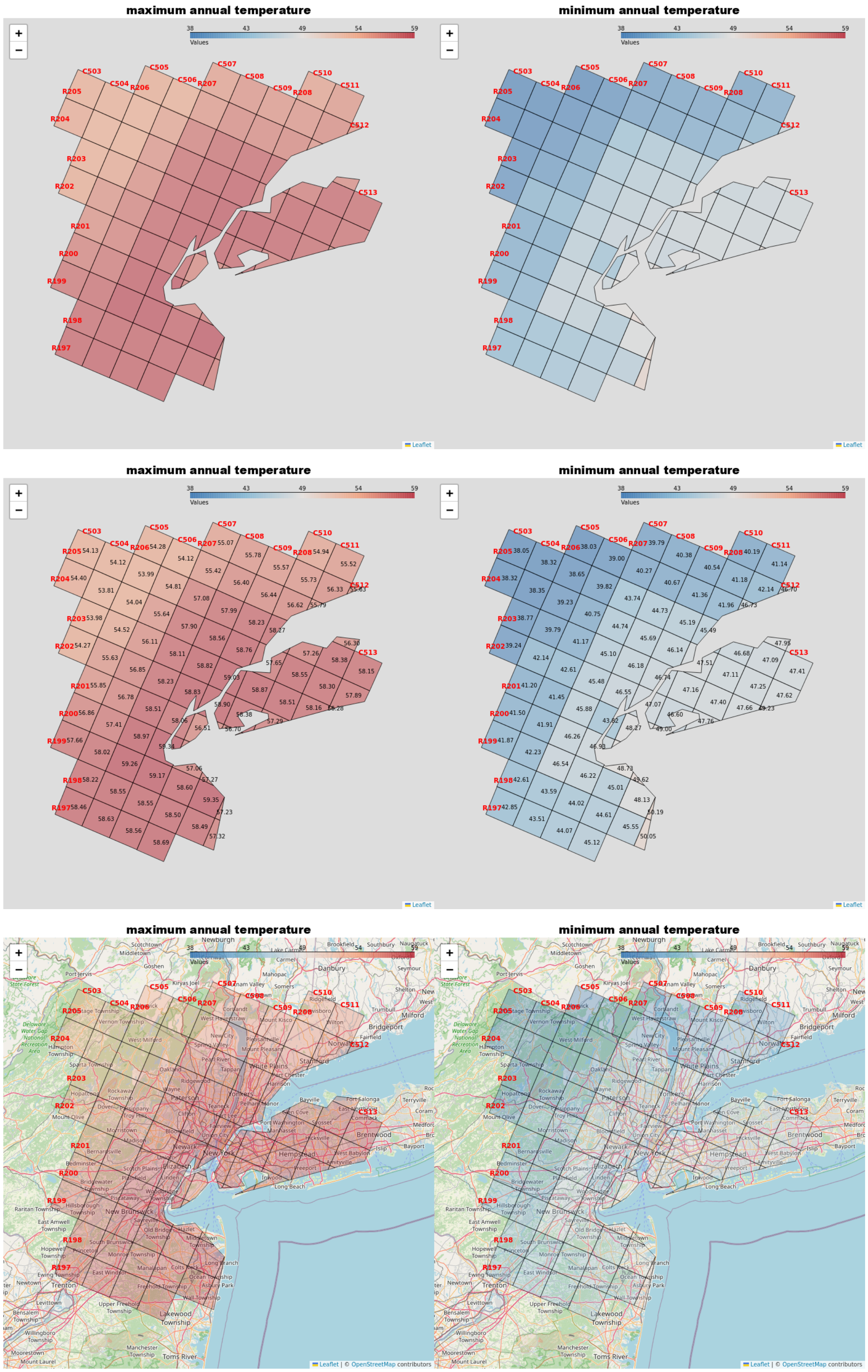


Figure 10. **Template 3:** What is the correlation between {climate_variable1} and {climate_variable2} in the {location1} during {time_frame1}? This example takes location1 = New York city, NY, climate_variable1 = maximum annual temperate, climate_variable2 = minimum annual temperate, and time_frame1 = historical period.

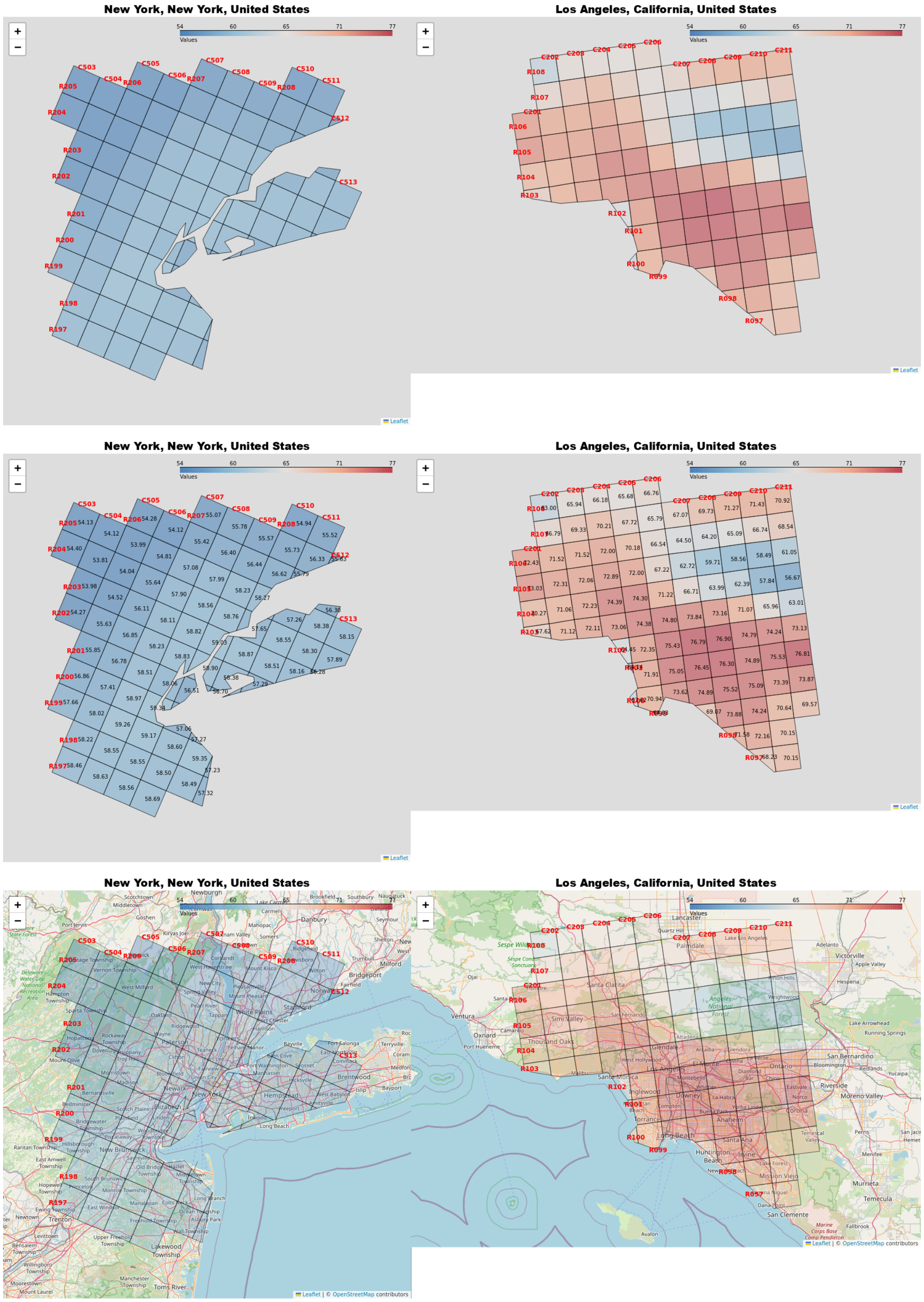


Figure 11. **Template 4:** How does {climate_variable1} compare between {location1} and {location2} during {time_frame1}? This example takes location1 = New York city, NY, location2 = Los Angeles, CA, climate_variable1 = maximum annual temperate, and time_frame1 = historical period.

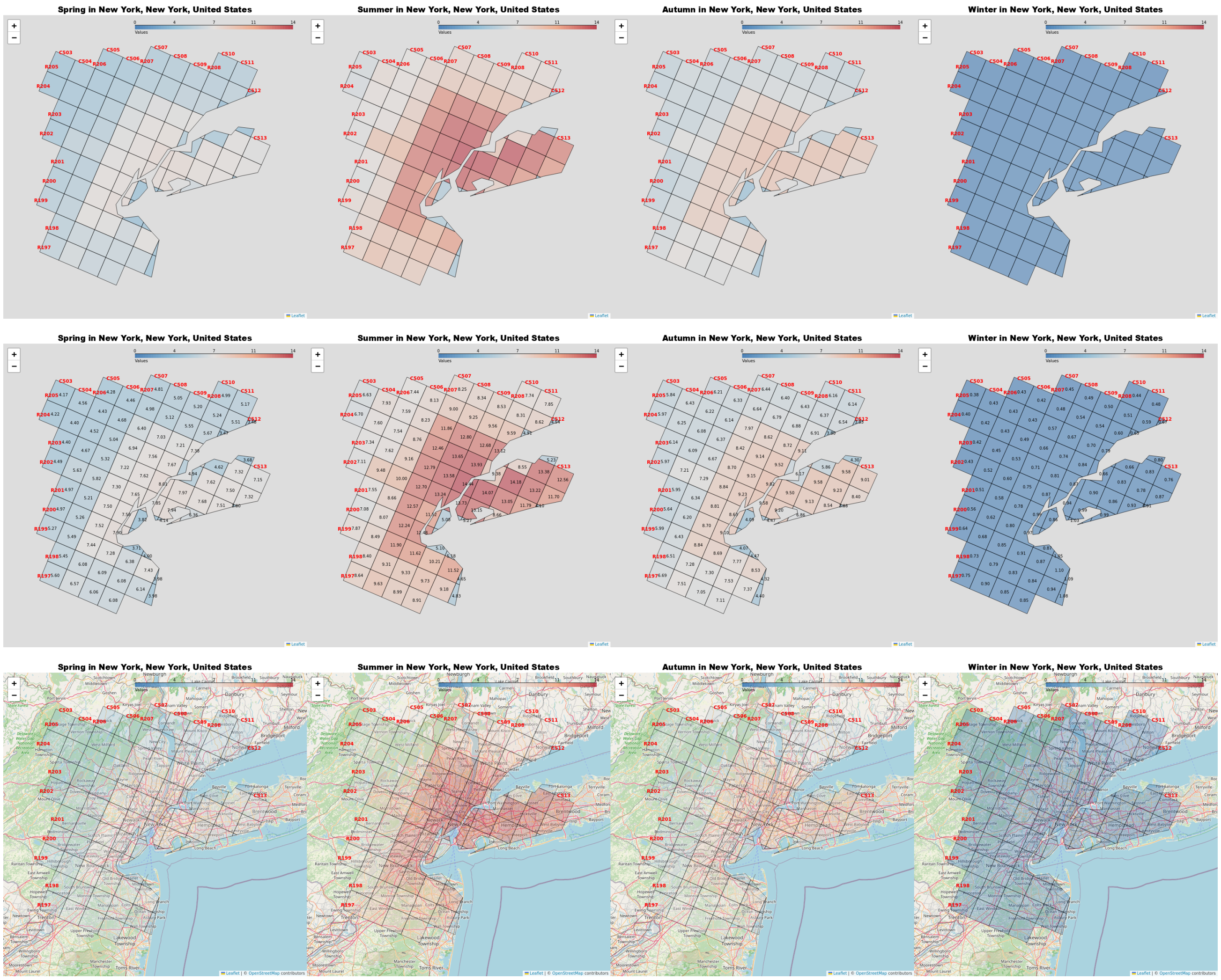


Figure 12. **Template 5**: What is the *seasonal* variation of {climate_variable1} in {location1} during {time_frame1}? Same data is used in **Template 6**: Which *season* in {time_frame1} saw the highest levels of {climate_variable1} in {location1}? This example takes location1 = New York city, NY, climate_variable1 = maximum annual temperate, and time_frame1 = historical period.

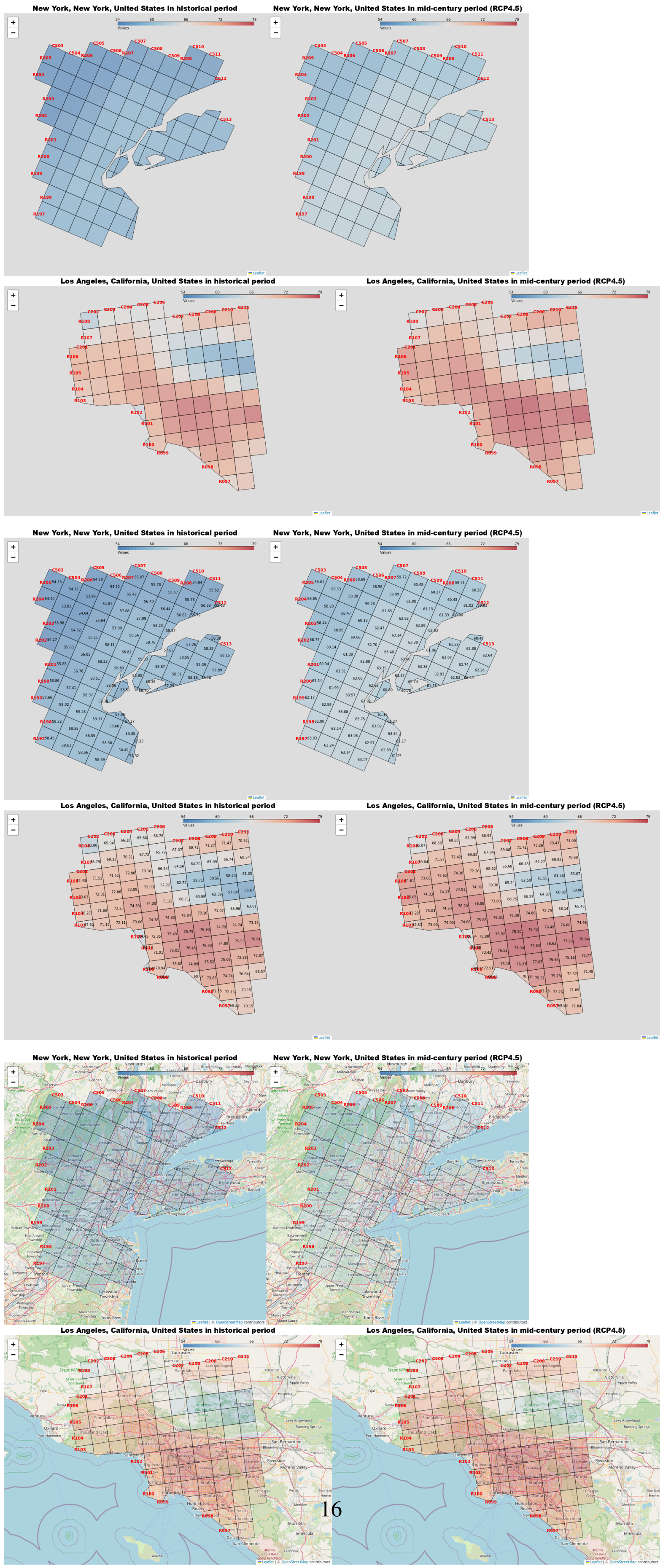


Figure 13. **Template 7.** Which of {location1} or {location2} experienced a greater change in {climate_variable1} throughout {time_frame1} and {time_frame2}? This example takes location1 = New York city, NY, location2 = Los Angeles, CA, climate_variable1

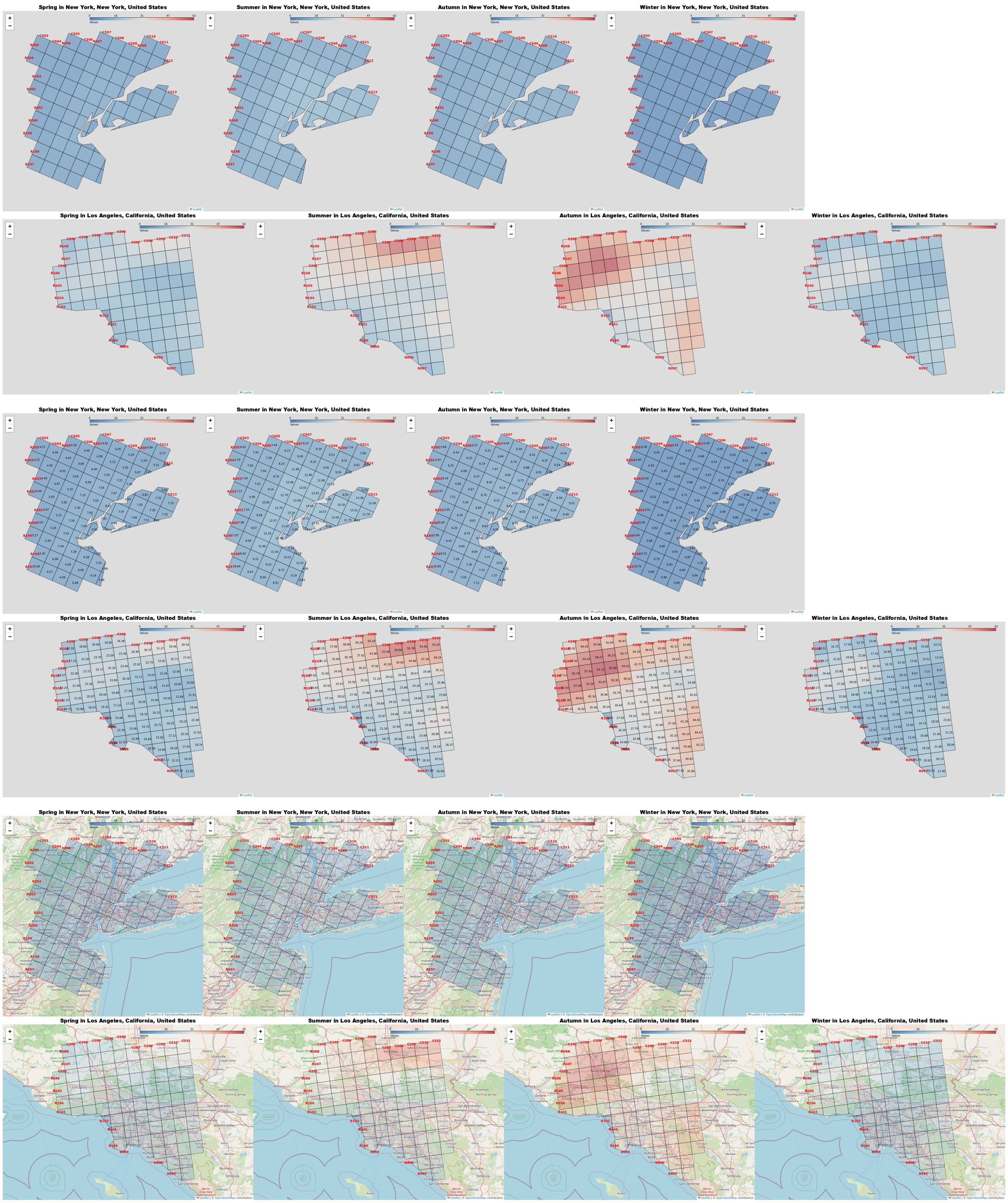


Figure 14. **Template 8.** How does the *seasonal* variation of {climate_variable1} in {location1} compare to that in {location2} for {time_frame1}? This example takes location1 = New York city, NY, climate_variable1 = maximum annual temperate, and time_frame1 = historical period.