Multiscale analysis framework for the Iowa Water-Energy-Food nexus

by

Weiquan Luo

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Agricultural and Biological Engineering

Program of Study Committee:
Amy L. Kaleita, Co-major Professor
Adina Howe, Co-major Professor
Philip M. Dixon

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this thesis. The Graduate College will ensure this thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2021

DEDICATION

This dissertation is dedicated to

Chenxi Zhang, whose support and encouragement
made the completion of this work possible

and

my parents, Yongde Luo and Huiying Chen, whose trust
strengthen me to overcome all challenges along this way.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to first thank my advisors, Amy L. Kaleita and Adina Howe, for the years of guidance professionally and academically throughout my master's program. They inspired me how my wrongness become science, or I would give up from my master's program. I would like to thank my committee members Dr. Philip M. Dixon for the time, and meaningful input.

I would like to thank my graduate student colleagues, Vishal Raul, Mingjun Ma, Paul Villanueva, Qing He, and Vitor Souza-Martins who have all made a positive impact on my experience throughout graduate school.

I would like to thank the Department of Agricultural and Biosystems Engineering at Iowa State University for providing the great support of this works, and for the faculty who are always encouraging and supportive.

I have been provided the opportunity to be successful in part from all of the people above and many more. I am grateful for all the support, encouragement, guidance, funding, and effort that all have provided to me during this process.

ABSTRACT

This research explored Iowa's Water-Energy-Food (WEF) resource system by modeling with Markov random field and analyzing with network analysis. With recognizing the gap between nexus modeling and communication in WEF resource management, the purpose of this research was to close the gap by proposing a framework of modeling and analyzing heterogeneous data in the WEF nexus discipline. The proposed framework aimed to discover interlinkage and characterize structural patterns between domains of Iowa's WEF resource system. The biophysical and economic data were collected from multiple sources and processed with a standardized data aggregation pipeline.

The first objective of this research was focused on the modeling method. The goal for this part of the research is to determine the technique to model Iowa's WEF resource system. The appropriate technique helps to identify the interlinkages between components with minimal subjectivity and closes the gap of communication via intuitive visualizations. We considered the model by the data availability and the capabilities of modeling and direct visualization at different scale. As a result, we proposed the method of coupling the data aggregation pipeline with the probabilistic graphical model that uses the same scheme to model and visualize a large-scale system at different scales.

The second objective of this research was focused on the characterization of Iowa's WEF resource system. The goal for this part of the research is to identify the interlinkages and structural patterns of the system across multiple spatiotemporal scales. The multiscale analysis grasped the characteristics of system at finer levels and connected the understanding of the behaviors of the overall system. Betweenness centrality, associativity coefficient, and degree distribution were applied to investigate the characteristics of the models across different scales.

As a result, we identify Iowa's WEF system is a free-scale, disassortative network that well-connected components would less likely connect to each other and more likely connect to the less-connected components. The analysis also suggested that the hydrologic responses was crucial in Iowa's WEF resource system.

CHAPTER 1. GENERAL INTRODUCTION

1.1 Background

A water-energy-food (WEF) resource system consists of entities from food, energy, and water subsystems that are connected by processes of mass and energy flow. The management of WEF resources has been identified as important for global sustainable development. As populations grow, the demand for WEF resources continuously increases causing the security of these resources to be at risk. Among 30 risks that have been identified to have significant global impact by the 2017 Global Risks Report, 26 of them impact at least one WEF resource sector and 9 of them impact on all three sectors (de Amorim et al., 2018).

To maintain a sustainable development against unstable environments, the management of these resources has shifted from a 'silo' approach to cross-sectoral approach in the past decades (Leck et al., 2015). While siloed management involves, cross-sectoral management is usually practiced by balancing the interests of all stakeholders to make decisions in response to one specific resource issue. Cross sectoral management has been recognized to be problematic, because it often solves one issue but often conflicts with other objectives. The Bonn 2011 Conference was convened with to help climate change. A result of this conference was the development of the notion of the WEF Nexus. This Nexus considers the three sectors in the WEF system are deeply intertwined, implying any changes in one resource portfolio could affect the others. The nexus approach is to integrate management and governance across sectors and scales (Hoff, 2011). The goal of a nexus approach is to maximize the gains by synergies and minimize the losses in trade-offs. Distinguishing from the cross-sectoral collaboration such as Integrated Water Resources Management (IWRM), the nexus approach emphasizes the interconnectedness of these three resource systems.

The WEF resource system in the Midwest region of the United States is a great example of synergies and trade-offs. Synergies and trade-offs are the two most important nexus concepts. During the 2000s energy crisis, many countries implemented a set of fuel subsidy policies to enforce a stable energy supply, such as the subsidies to support the biofuel industry in the United States. As the direct result, the subsidies shielded consumers from higher oil prices ("Crude Measures," 2008). As a trade-off, the biofuel industry had a huge demand on fuel crops and resulted in a higher price of agricultural commodities (Mitchell, 2008). The increasing world food prices later created the 2007–2008 world food price crisis, which caused political and economic instability and social unrest. Moreover, the subsidies policy changed the agricultural practices. These practices had a synergistic effect on the water quality issues, one effect being eutrophication. As the food prices became higher, farmers chased higher crop yields by installing tile drainage systems and applying higher doses of fertilizer. When fertilizer is not fully utilized by the growing plant, they leach through the soil over time and increase the levels of nitrogen and phosphorus in groundwater, which causes eutrophication of a water body. Improper water management of the tile drainage system intensifies the leaching process of the fertilizer. These political, economic and environmental consequences of subsidies policy, although unintended, are long-term and costly to fix. To minimize such consequences, we need to understand the behaviors of the WEF resource system before making decisions.

1.2 Existing Methods and Challenge

To understand correlation within and between the biophysical system and human system, much research has been done on nexus modeling We categorized the existing nexus modeling method into two approaches distinguished by the input data, specifically the biophysical data and the opinion data. Both approaches focus on quantifying the strength of correlation but in different ways. One approach uses biophysical data to simulate metrics with mathematical models or biophysical models. This approach includes most of the current available methods, such as life cycle assessment, input output analysis, the LEAP (SEI, 2013), the WEAP (SEI, 2014), MuSIASEM (FAO, 2013), CLEWS (KTH, 2013), and WEF Nexus Tool 2.0 (Daher & Mohtar, 2015). This approach is rarely designed to find new relationship. The relationship are often predefined by the underlying models. Their results are objective but restricted by the available data and the underlying model for simulations. Another approach encodes WEF stakeholders' opinions from interviews the interlinkage of a social network model that consist of node and arc to represent the correlation between activities in the WEF system (White et al., 2017). The resulting graph is straightforward and easily interpreted, but stakeholders' opinions can sometimes be biased. Beyond the pros and cons, these available modeling approaches have at least one of the following issues: modeling methodology, rescaling capability, difficulty of communication.

The first issue is modeling methodology. Some methods are originally the gold standard in one subject area and in fact violate the nexus perspective. The analysis first performs for the original subject area, then it is modified to include variables from other WEF domains. The modification is subjective to analysts' interests, so the analysis may not lead to a convincing result that fairly evaluates the importance of each resource subsystem.

The second issue is rescaling capability. Some methods are not flexible in handling heterogeneous data that has different spatiotemporal resolutions. The analysts must decide a scale of resolution, which is often defined by the physical model parameters or by the most common scale among the available data. The performance of these analyses is limited at one scale by aggregating or converting heterogeneous data to an identical resolution. The scales where the correlative relationship may occur are not clear, but the most common scale among data is chosen without detailed justification. Without rescaling capability, the analysis at multiscale cannot be done.

The third issue of the existing modeling techniques is the difficulty of communication. The WEF system is a multidisciplinary subject. To communicate this subject requires a wide range of knowledge, but most stakeholders' expertise are limited to their own fields. Any jargon or discipline-specific terminology would increase the difficulty of communication between different groups of people. So, communication is a huge gap between analytical results and decision making. Above all, the restrictions and issues make most existing approaches difficult to apply in real-world scenarios.

1.3 Objectives

In this study, the objective was to develop a model for a WEF system which can intuitively integrate and describe correlations at multiple scales of spatiotemporal resolutions and close the gap of communication. As the common practice in the nexus studies, all WEF variables were maintained to be equally important in data integration. We specifically define the pairwise relation between two variables using the correlation while conditioning on the other variables. By representing the dependencies as a link on probabilistic graphical models (PGM), we can understand the system dynamics of a WEF resource system at a particular scale. The second objective is to understand relationship and structural patterns of a WEF resource system at multiscale. We identified the WEF entities and domains that are more important in the system and explain why these entities and domains are outstanding.

1.4 References

Crude measures. (2008, May 29). *The Economist*. https://www.economist.com/briefing/2008/05/29/crude-measures

Daher, B. T., & Mohtar, R. H. (2015). Water–energy–food (WEF) Nexus Tool 2.0: Guiding integrative resource planning and decision-making. *Water International*, *40*(5–6), 748–771. https://doi.org/10.1080/02508060.2015.1074148

de Amorim, W. S., Valduga, I. B., Ribeiro, J. M. P., Williamson, V. G., Krauser, G. E., Magtoto, M. K., & de Andrade Guerra, J. B. S. O. (2018). The nexus between water, energy, and food in the context of the global risks: An analysis of the interactions between food, water, and energy security. *Environmental Impact Assessment Review*, 72, 1–11. https://doi.org/10.1016/j.eiar.2018.05.002

FAO (2013). An innovative accounting framework for the food-energy-water nexus Retrieved from http://www.fao.org/docrep/019/i3468e/i3468e.pdf

Hoff, H. (2011). Understanding the Nexus. Background Paper for the Bonn2011 Conference: The Water, Energy and Food Security Nexus. *Stockholm Environment Institute, Stockholm*.

KTH (2013). CLEWs – climate, land, energy and water strategies to navigate the nexus. Last accessed on July 2, 2014. Retrieved from http://www.kth.se/en/itm/inst/energiteknik/forskning/desa/researchareas/clews-climate-land-energy-and-water-strategies-to-navigate-the-nexus-1.432255

Leck, H., Conway, D., Bradshaw, M., & Rees, J. (2015). Tracing the Water–Energy–Food Nexus: Description, Theory and Practice. *Geography Compass*, *9*(8), 445–460. https://doi.org/10.1111/gec3.12222

Martin-Nagle, Renee; Elizabeth Howard; Alyssa Wiltse y David Duncan (2012), Bonn 2011 Conference "The Water, Energy and Food Security Nexus" — Solutions for the Green Economy, 16-18 November 2011. *Conference Synopsis, Federal Ministry for the Environment, Nature Conservation and Nuclear Safety (BMU), Federal Ministry for Economic Cooperation and Development (BMZ), OOSKAnews, Inc.*

Mitchell, D. (2008). A Note on Rising Food Prices (SSRN Scholarly Paper ID 1233058). *Social Science Research Network*. https://papers.ssrn.com/abstract=1233058

Part 1: Global Risks 2014: Understanding Systemic Risks in a Changing Global Environment. (n.d.). *Global Risks 2014*. Retrieved July 12, 2019, from http://wef.ch/GJKWJg

SEI (2013). Long range energy alternatives planning system. Tool found on: http://sei-us.org/software/leap

SEI (2014). Water evaluation and planning. Tool found on: http://www.weap21.org/index.asp?action=200

White, D. D., Jones, J. L., Maciejewski, R., Aggarwal, R., & Mascaro, G. (2017). Stakeholder Analysis for the Food-Energy-Water Nexus in Phoenix, Arizona: Implications for Nexus Governance. *Sustainability*, *9*(12), 2204. https://doi.org/10.3390/su9122204

CHAPTER 2. METHOD

The empirical data in a large-scale WEF modeling study (WEF data) includes variables that represent different aspects of the WEF system components. WEF data is generally heterogeneous due to the high variability of the spatiotemporal resolution. The WEF data related to different resource sectors are often collected by different agencies with different sampling frequencies at different spatial scales. Many modeling techniques require homogeneous data, so it is necessary to aggregate the WEF data to an identical resolution.

However, aggregating heterogeneous data is challenging because the unique identifiers between records of two different datasets often do not exist (Wang, 2017). For example, a WEF data may include corn yield and precipitation. Corn yield is usually reported at a county-level or watershed-level once or twice per year; whereas precipitation is recorded on the hourly-basis for one particular weather station. Neither unique spatial nor temporal identifiers between corn yield and precipitation exist to allow them to match and merge. Therefore, the rescaling method has to be considered before to incorporating heterogeneous variable into a model. For example, variables can be aggregate by time period and region, then each datapoint in the dataset contains summarized information of those variables at a time period and a region.

The proposed method consists of two steps: data aggregation to multiscale and the probabilistic graphical model. As the first step, the data aggregation systematically integrates the heterogeneous dataset to designated identical scales by considering the data resolution, the designation of scales, and the appropriate upscaling arithmetic. The second step is to use a probabilistic graphical model to identify the strength of correlation between variables. By considering the multiscale modeling capability, the availability of analysis technique, and nexus perspective of the modeling technique, we select Markov random fields to model the WEF resource system.

2.1 Data Aggregation to Multiscale

The existing nexus studies took a traditional approach that models focus on one scale. The one scale is always a macroscopic level because the modeling is more feasibly supported by the data availability and the empirical evidence at this level of resolution. In contrast, although the microscale modeling is rarely feasible and not able to offer efficient information, the microscale models can provide more accurate detail once the modeling is successful. One simple way to balance the efficiency and accuracy is to take a multiscale approach to understand the system at both macroscopic and microscopic levels.

The proposed method is to fit the model with the WEF resource system at multiscale. Compared to the modeling at one scale, multiscale modeling is an alternative approach in which multiple models at different scales describe a system simultaneously. The different models capture the interlinkages on the identical WEF system. Each interlinkage has various levels of strength across different resolutions. Some linkages are obvious at microscopic level, such as heat transfer; where some linkage develops through a process integrated over time or space, for example, continuous dry days causes death of crop. By screening interlinkages at both microscopic and macroscopic level, we expect to illustrate a holistic view of a WEF system with compromise between accuracy and efficiency.

The data preparation is the key to the successful multiscale modeling of a WEF system. The preparation of upscaling original data at different resolutions to an identical scale is a delicate process of data integration. There are three essential elements in the process of data integration including the data resolution, the designation of scales, upscaling arithmetic operation. Each variable in the original data needs to be collected at its highest available resolution. It allows the interlinkages to be discovered at the finest levels. Given the high-resolution data, the multiscale modeling is performed at the scales of different temporal and spatial resolutions. Specifically, the candidate of temporal scales can be daily-level, monthly-level, quarter-level, and yearly level; the candidates of spatial scales can be either the classification of hydrologic units or the administration division whichever way the data is available under the scales. With the data and designated scales, the appropriate upscaling arithmetic operation is crucial to maintain their essential meaning of each variable at a lower resolution. The result of the upscaling operation is a statistic, such as mean, mode, median, maxima, and minima.

2.2 Probabilistic Graphical Model

After data integration, the upscaled data at identical spatiotemporal resolution is fitted with the probabilistic graphical model (PGM) in which a graph expresses the conditional dependence structure between random variables. Compared to the methods of the existing physical models, the PGM has advantages to the three issues of nexus modeling in methodology, scalability, and communication. First, PGM is specialized in discovering connection between variable (Koller & Friedman, 2009). PGM encodes the dependency relations among variables with joint probability distributions. The joint distributions hold the information of statistical dependence, which is the potential interlinkage in a WEF system. Secondly, PGM is a probabilistic model that distinguishes from simulation models with assumption on physical models. It generates the joint probability distribution when a sufficient amount of complete data is given. PGM does not build upon any physical model. Without any assumptions of variables on physical units, the modeling and analysis at the various scales of spatiotemporal resolution are possible with PGM. Thirdly, the result of a PGM is represented as a network consisting of vertices referring to variables and links referring to the dependency relations among variables. Compared to the simulation model resulting in numerical values, PGM uses an intuitive representation to illustrate interlinkages and provides a holistic view of a WEF system. Therefore, PGM is an appropriate method of multiscale analysis to investigate the correlative relation and for easy communication.

The network analysis can be applied to the network resulting from PGM to understand the pattern of the structure. As the generic techniques of network analysis, the degree and centrality analysis have been used in the WEF studies. The degree is a descriptive measurement of the number of connections of a vertex to others. The assortativity coefficient derived from the distribution of the degree describes the similarity in degree among all vertices in a network structure. By measuring the model with centrality of each subsystem and assortativity of the network structures, PGM with network analysis can advance the understanding of the pattern of the interlinkage in a WEF system. For example, the assortativity coefficient of the interstate virtual water trade network (VWTN) reported the disassortative behavior that the states with a large number of connections are connected to the states with fewer connections in the United States (Vora et al., 2017). The assortativity coefficient of interstate VWTN also supports the fact that the majority of states participate in the food trade, and only some states actively engage in energy trade (Mahjabin et al., 2020). For instance, the centrality measurement compares the importance between vertices. By evaluating the centrality of vertices grouped by subsystem, the synergy within subsystems can also be explored and analyzed (Li et al., 2019).

The undirected form of PGM is appropriate to illustrate a WEF system with a nexus perspective. Based on whether network structures are directed or not, PGM can be mainly divided into two branches, Bayesian networks (BN) and Markov random fields (MRF). The network structure of BN is a directed acyclic graph, whereas the network structure of MRF is an undirected graph. Within the nexus perspective, reciprocal interlinkages are common in a WEF system, such as the circular agriculture between corn, feed and animal production. The animal agriculture supplies manure as the fertilizer for corn production; the corn is used as animal feed in animal production. The reciprocal relations can be illustrated in a directed graph. However, a directed graph generally requires no cyclic feature. The acyclicity limits the directed graph from the representation of such reciprocal relations. Although the undirected graph cannot show any directions of interlinkages, they appropriately illustrate a WEF system as the network structure.

2.3 References

Li, G., Wang, Y., & Li, Y. (2019). Synergies within the Water-Energy-Food Nexus to Support the Integrated Urban Resources Governance. *Water*, 11(11), 2365. https://doi.org/10.3390/w11112365

Mahjabin, T., Mejia, A., Blumsack, S., & Grady, C. (2020). Integrating embedded resources and network analysis to understand food-energy-water nexus in the US. *Science of The Total Environment*, 709, 136153. https://doi.org/10.1016/j.scitotenv.2019.136153

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. MIT Press.

Vora, N., Shah, A., Bilec, M. M., & Khanna, V. (2017). Food–Energy–Water Nexus: Quantifying Embodied Energy and GHG Emissions from Irrigation through Virtual Water Transfers in Food Trade. *ACS Sustainable Chemistry & Engineering*, *5*(3), 2119–2128. https://doi.org/10.1021/acssuschemeng.6b02122

Wang, L. (2017). Heterogeneous Data and Big Data Analytics. *Automatic Control and Information Sciences*, *3*(1), 8–15. https://doi.org/10.12691/acis-3-1-3

CHAPTER 3. INVESTIGATE LINKAGES IN IOWA WATER-ENERGY-FOOD NEXUS USING SPARSE MARKOV RANDOM FIELD AT MULTISCALE

Weiquan Luo, Amy L. Kaleita, Adina Howe
Department of Agricultural and Biosystems Engineering, Iowa State University

3.1 Abstract

The water, food, and energy (WFE) are three sectors that are necessary for sustainable development, but the relation among the WEF security sectors are complicate to be understood. We developed a multiscale analysis framework to understand Iowa's WFE systems. The framework consisted of data integration pipeline, probabilistic graphical modeling, network analysis. The framework was capable in homogenizing data with different scales, visualizing relationship between sectors, and characterizing structural patterns of the system. The result shown that the network model of Iowa's WEF system was disassortative. More importantly, we identified that the hydrologic system played an important role in connecting other systems in the Iowa's WEF system.

3.2 Introduction

Iowa is a unique state in the Midwestern United States because of the central role of agriculture in its economy. In 2012, over 85% of the land was used for agriculture. Iowa has the top national ranking in corn production (2.5 billion bushels), grain storage capacity (3.6 billion bushels), and hog (23.5 million) and egg (16.2 billion) production, and it is ranked second in soybean production (564.8 million bushels) (USDA, n.d.). Manufactural and energy production industries rely on the raw materials from this large agricultural system. For instance, 39% of the corn grown in Iowa creates nearly 30% of all American ethanol; however, the growth of biofuel industries raised the concern of diverting farmland for biofuels production to the detriment of the food supply (Iowa Corn Growers Association, 2016). To provide the feed and fertilizer for the massive agricultural production, Iowa has developed a sustainable agricultural economy. Livestock consumes about 21% of Iowa grown corn (Iowa Corn Growers Association, n.d.), and the livestock production supplies about 25% of the cropland fertilizer needs (Iowa Pork Producers Association, 2019).

While food and energy resource systems maintain a subtle balance, water quality indicates the large-scale environmental status in terms of quality degradation. Water management practices have made substantial progress in many ways. For example, through the use of no-till farming management, contour farming, terracing, ground cover, and wetland conservation. Still, the water quality in Iowa was directly or indirectly influenced by the process of daily farming practices and long-term land-use planning. Nutrient pollution is the major water quality issue, which eutrophication and soil erosion often accompany. The nitrogenized water causes severe algal blooms in the local waterways and along the Mississippi River to the Gulf of Mexico. We could use some biophysical model to understand the impact of one on another, but it is difficult due to the complexity of the water-energy-food (WEF) resource system.

After the Bonn 2011 Nexus Conference, challenges in resource management of the WEF resource system have increased the global interests in nexus-based frameworks. The concept of the WEF security nexus describes how actions in any one particular resource system often have effects in other resource systems. To fill the gap between the nexus theory and its applications, some methods have been developed for resource management optimization. Currently, the integrated modeling approach is popular for nexus modeling. The global Climate, Land, Energy, and Water Systems model is an integrated framework to evaluate resource security and policy impacts but does not comprise separate water or climate modules (KTH, 2013). WEF Nexus Tool 2.0 is an online tool that evaluates resource requirements and sustainability indices at different proposed scenarios (Daher & Mohtar, 2015). The Water, Energy and Food security nexus Optimization model (WEFO) is a multi-period optimization model that is capable of addressing the temporal features of a WEF system with existing socioeconomic and environmental constraints (Zhang & Vesselinov, 2017). It measures complicated casual relationships by reducing the multidimensional and codependent uncertainties using model simulation.

A major challenge for creating a successful model is how to integrate high-dimensional data collected from different scales into a single model. Most models consider one spatiotemporal resolution, particularly a regional-annual scale. The setup of a single scale limits the model from investigating the relations of variables at different scales or across scales. Without knowledge of the exact scale for all relations, multiple single-scale analyses could illustrate a more complete picture of the system.

The single-scale modeling is conventional in nexus studies, but the multiscale analysis is rarely explored and still needs further advancements (Endo et al., 2020). Importantly, collecting sufficient data for a higher resolution analysis is recognized to be a challenge (Khan et al., 2018).] In this study, we proposed a methodological solution for multiscale analysis by constructing multiple models at different scales to study the interlinkages in the Iowa's WEF system. A two-step method was applied to the nexus. The first step was a data integration workflow that upscaled the data to lower designated scales. The second step was to couple the nexus into a probabilistic graphical model, specifically Markov random field (MRF). MRF was selected because of its ability to represent generic nexus features of undirected cyclic dependencies.

3.3 Study Area

In this study, we used Iowa as the study area. Iowa is a rural, agriculture-intensive area located in the Midwest. Farms make up more than 85 percent of Iowa's land, where 70 percent is for cultivated crop (National Agriculture in the Classroom, 2020). Iowa has little to no need for irrigation, because it generally has plenty of rainfall, especially in the summer. Importantly, the hydrologic process has deterministic effect on the exchange of matter between land. As the result, the spatial feature of crop planting and seasonality of water supply make up a large and dense hydrologic network that connect all components in this WEF system. Besides the biophysical system, the human activity is an essential part of a WEF system (Kling et al., 2017). The farmland and human environment are two primary sources of wastewater. The farming activities, particularly animal waste management and fertilizer application, affect water quality and agriculture production. The human activities, such as food manufactures and biorefinery industries, demand the local raw material to minimize transportation costs. The agriculture-intensive WEF system coupled with narrow human activities results in a unique WEF nexus for Iowa.

3.4 Data Collection and Scale

Multiscale analysis for a large area requires a massive amount of data. In this study, we collected data relative to the biophysical and human systems to outline the fundamental nexus of Iowa, including hydrology (i.e., weather and water), agriculture, energy, economy, and environmental events counting, shown in Appendix A. The weather, water, agriculture, and energy data were typical in nexus studies. The economic data reflected on the human componence in the system. The data of environmental events counting were closely related to the issues concerned by the Iowa public.

Space and time were two dimensions considered in this multiscale study. In the public-available data, the periods of daily, monthly, and yearly were identified as the three typical temporal scales. Point-level, county-level, and state-level data were the three typical spatial scales. We defined the level of scale by the number of the data point per unit area and period. Therefore, a large scale was made up of high-resolution data, such as Point-level and daily-level; whereas a small scale was made up of low-resolution data, such as state-level and yearly-level. To maximize the capacity of the multiscale analysis, we collected the data at the smallest available scales.

For detailing the multiscale analysis, the agricultural district (or AgDistrict) was added as an intermediate spatial scale. AgDistricts were designed by state law to encourage the use of land for farming supervised by the county board. The 99 counties in Iowa were separated into nine AgDistricts, of which each consisted of about nine to twelve counties, shown in Figure 1. The AgDistricts afford legal protections and some tax benefits for viable agricultural land (Iowa Code, 2019). Aggregating biophysical variables by AgDistricts was used to augment the difference of the administrative activities by regions. Moreover, economic data were only surveyed or reported by administrative division. Therefore, this multiscale study consist of nine combinations of three spatial scales (county, AgDistricts, and state) cross three temporal scales (daily, monthly, yearly).

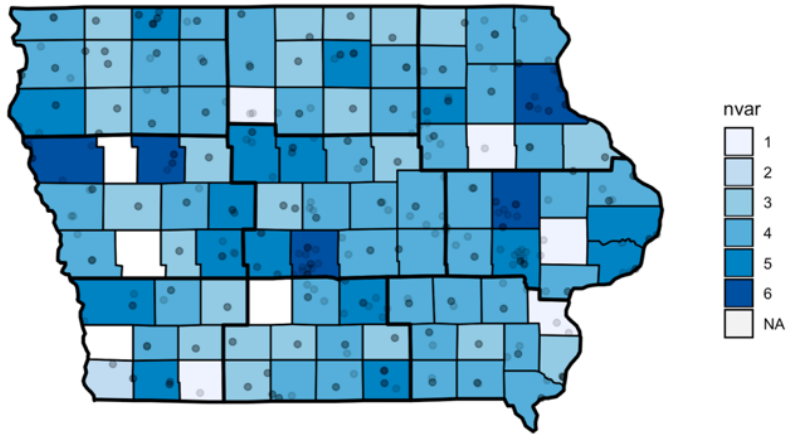

Figure 1: A county-level map of Iowa State with the boundary of nine agriculture districts, where counties are colored by the number of available hydrology-relative variables.

3.5 Method

A two-step method was applied to the nexus in the Iowa region shown in Figure 2. The first step was to integrate the heterogeneous data to designated scales of space and time where data were condensed from a couple of gigabits to a few megabits. It included upscaling, variable selection, and transformation to result in a complete dataset with appropriate data distribution required by the model. The second step was to fit the MRF model with the integrated data.

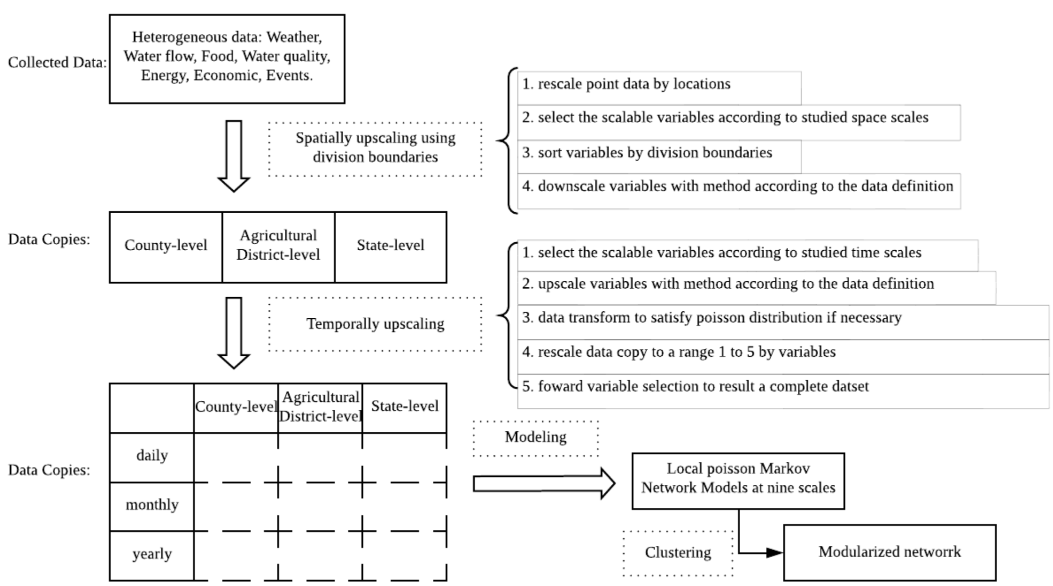

Figure 2: Workflow of the WEF nexus modeling with data integration through upscaling and Local Poisson Markov Network modeling.

3.5.1 Data Integration to Resolve Heterogeneity

The single scale was favored in most nexus research because homogenizing the collected data to larger area and to yearly-scale make analysis easier. For instance, data relative to weather, water, and environmental events counting (e.g., high winds, flood, droughts, fishkill) were originally collected as daily point data from water sampling sites or weather stations. In contrast, the data related to economy and energy were commonly collected every month. Further, data related to food, energy and economy were mostly available in low spatial resolution. For example, the employment rate was reported monthly by the U.S. Bureau of Labor Statistics due to the difficulty of survey; corn yielded once per year in Iowa due to local planning seasons. The data collected from different locations and reported on different temporal scales could not be compared, so they needed to be integrated to identical resolutions to be analyzed. The data integration consisted of two major processes, including upscaling and variable selection.

3.5.1.1 Upscaling

The upscaling was the processing that the data were upscaled to lower resolution, for example, a point data or county-daily variable was aggregated to a state-yearly basis. Each variable had different upscaling methods. The upscale methods, including averaging, minimizing, or maximizing, was applied to statistical data, such as physical measurement and the periodical average of price. The summation was applied to the counting data such as event counting, account value, and inventory. Additionally, some collected data were already at the smallest resolution of space or time or both, so they were unscalable. The upscale method was applied to them according to the above categories at two dimensions, respectively. As a result, the scale heterogeneity of the original data was resolved by the upscaling process.

3.5.1.2 Variable selection

After the original data were upscaled to a designated resolution, a forward selection was applied to form nine dataset at different resolutions, respectively. Since the water and weather data were point data and carried the highest resolution information, variable selection process started with the inner joined dataset of water and weather data. The rest of the variables engaged in a join-select-clean procedure repeatedly till reach a stopping condition. Each of candidate variables was tried to inner join with the initial dataset, respectively, to result many joined dataset. Among all these candidate variables, the variable with resulting in the highest percentage of datapoint remaining in the joined dataset was selected to be kept in the initial dataset. The combined dataset was probably incomplete due to missing values in some data points; these data points with missing values were removed to form a complete joined dataset. Therefore, as more variables were added to the joined dataset, fewer data points remained due to the removal of missing values. The selection process repeated until reaching a threshold that was designed to include as many variables as possible and retained enough datapoints for modeling. We decided to use 80% as the threshold. When a variable shrank the size of previous datasets more than 20%, this variable was excluded. Noticeably, this stepwise algorithm is a greedy method, so it may not find the best dataset that contain the most information. By reaching the threshold in the variable selection process, data integration was completed, and joined dataset is ready for modeling.

3.5.2 Sparse Exponential Markov Random Fields Formulation

For modeling these datasets, we used Markov random fields (MRF), also called the Markov network or undirected graphical model. This model is a family of probabilistic graphical models that compactly represents the dependencies among variables. For this specific task, the random variables are the collected data to outline a WEF system. Suppose $X = \{X_1, ..., Xp\}$ is a random vector with P dimensions corresponding to the P variables. The undirected graph G = (V, E) has vertices $V = \{X_1, ..., Xp\}$ and a set of edges E corresponding to pairs of vertices. In the context of nexus modeling, the undirected graphs describe the probabilistic behavior in a WEF system, specifically, the correlation for each variable at spatiotemporal resolution. Distinguished from the stochastic model and spatial model, such as Markov chain and Ising model, the resulting network models does not consider neighboring correlation. For example, the spatial correlations between adjacent regions and consecutive time periods.

3.5.2.1 Probability Distributions Assumption

In general, the structural learning algorithm of probabilistic graphical models does not customize distribution assumption regarding each variable. MRFs have traditionally been applied to the Gaussian or binary variables in the field of computer vision or bioinformatics. By extending from the Gaussian distributions to the exponential family, the MRF can gain the capacity of modeling data with count data. It is the reason why the probabilistic graphical models are usually applied to the dataset with homogeneous data type, such as RNA-seq count. However, this study covered multiple field of study. Therefore, it is unavoidable to have a dataset including discrete and continuous variables with overdispersed and skewed distributions. Due to the diversity of the distributions, the traditional MRF is not appropriate for our dataset. Developing a new structural learning algorithm is not the major objective of this study. To balance the performance of the model fitting various distributions of all variables and their aggregation, we discretized each variable to five level with equal interval width. With understanding the increasing overdispersion after upscaling could offset some skewness for some variable, the right skewed remain. We assume Poisson distribution for variables, because it has right skewed property. It contains only one parameter, which is less computational expensive by trading off accuracy. As a compromise solution, we used an MRF specified for local-Poisson distribution in the exponential family to learn the network structure, with the assumption of univariate distributions at each variable. After being upscaled to designated resolutions, the data is fitted with MRF to result in nine models corresponding to the nine spatiotemporal scales.

3.5.2.2 Sparse model for high-dimensional data

A network model is sparse when there are only a few connections for each vertex. The sparse network model simplifies the complex relations of high-dimensional data by only emphasizing strong connections. The standard techniques of MRF in evaluating global network, including K-fold cross-validation, Akaike information criterion (AIC), and Bayesian information criterion (BIC), tend to select overly dense graphs in high dimensions, where the number of edges is close to the maximal number of edges (Liu et al., 2010). The overwhelming number of edges causes the enormous difficulty in understanding the network. Therefore, the network needs to be as sparse as possible so that massive weak connections do not bury important connections. To infer a sparse structure, we used the XMRF package in the statistical software environment R to infer the network structure. The package provided the functionality of fitting the exponential family MRF to data through optimization on each node by Newton's method (Wan et al., 2016). The algorithm had strong theoretical guarantees for sparsity through penalizing node-conditional likelihood estimation. Instead of the standard techniques using the score-based (e.g., AIC, BIC, etc.) or test-based method, the algorithm used a stability-based method to regularization selection for high dimensional graphical models, or namely sparsity. (Yang et al., 2012) To determine the network sparsity, we apply the stability-based method, which retained network edges that are estimated in more than 95% of the 1000 bootstrap repetitions. As the result, the weights of edges in the networks would end up close to zero for no connection or one for connection.

3.6 Result and Discussion

3.6.1 Feature Dimension and Sampling Density

We first evaluated the ability to upscale the original dataset. The original dataset contains 31 variables with different resolutions. Some variables were removed in order to combine these variables. We evaluated two main features of combined datasets, the total feature dimensions and sampling density. Our goal was to evaluate how much information was reserved after upscaling.

Suppose a feature space represents all collected data in this study, the feature dimension referred to the number of variables. For example, the combined dataset on daily-county scale had six variables, then the feature dimension was six. As the result of data integration, the feature dimension become greater from daily scales to yearly scales, as shown in the upper graph of Figure 3A. The feature dimension increased along temporal scale shown that more variables were available at the yearly scale than monthly scale and daily scale. While the obvious difference occurred across temporal scale, the feature dimensions were not much different across spatial scales, shown in the upper graph of Figure 3B, which suggested that models at the same temporal scale are likely to have similar variables.

The sampling density was represented by the number of the integrated data points sampled in the feature space. The sampling density decreased greater along temporal scale than spatial scale from larger to smaller scales, shown in the lower graphs of Figure 3. The decrease of sampling density came from two sources, including the scale of resolution and the removal of incomplete data point during data integration.

Determined by the scale of a spatiotemporal resolution, the maximum amount of data points always decreased from higher resolution to lower resolution and was greater along temporal scale than along spatial scale depending on the designated scale. For example, a 10-year data integrated to county-yearly level could have maximum 11880 data points, which was the multiplication of 10-year, 12-month, and 99-county. With a perfect dataset that all variable were available at county-daily scale, the maximum amount of data point in the integrated data decreased by factors of 12 from monthly to yearly and 9 from AgDistrict to state level, whereas the factors were 365 from daily to monthly, and 99 from county to AgDistrict level.

The default decreasing pattern appeared along temporal scale, but not along the spatial scale. If complete data was available at most counties, the amount of data point was expected to be greater in county-level than in Agdistrict-level. The result shown in the lower plot of Figure 3B implied the daily data is heavily concentrated at a few counties. In fact, only 5 out of 99 counties had complete data available at daily-county level due to data avalibility of water temperature. The rest of the counties were excluded from the integrated dataset due to missing information. When the variable of water temperature was removed, the model has ability of demonstrating the overall view of the system. By analyzing data availability using feature dimensions and sampling density, we expected the models at the same temporal scale were likely to be more similar than at the same spatial scale due to the data availability.

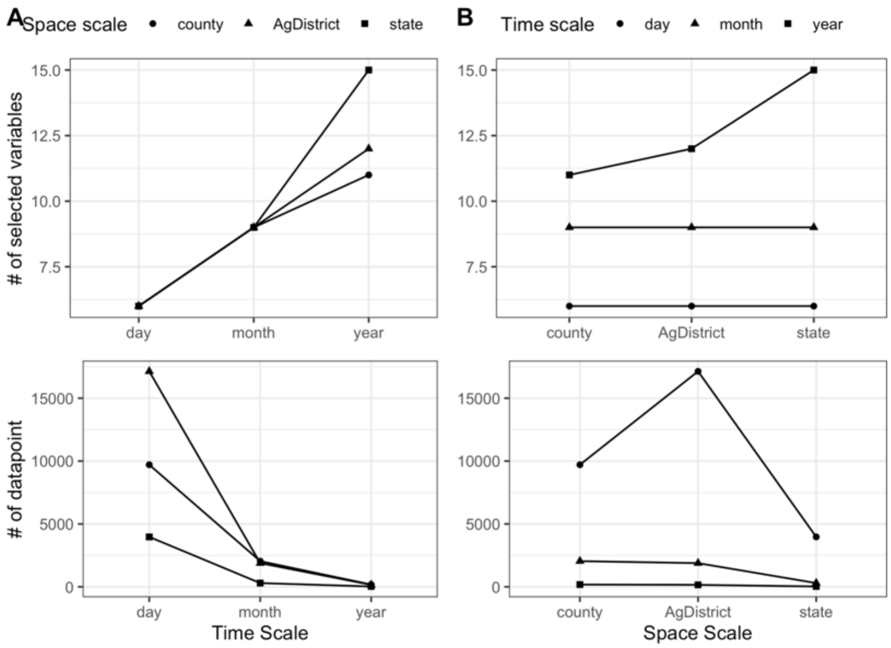

Figure 3: The comparison of integrated datasets between spatial scale across temporal scale (A) and temporal scale across spatial scale (B) shows that the feature dimension and sampling density vary in temporal scale and spatial scale.

3.6.2 Model at Multiple Scales

We obtained nine models at multiple scales by fitting the network models to the combined datasets. Each model represented the connection between variables at the corresponding resolution, as shown in Figure 4. The nine models as a whole could provide an intuitive representation of the Iowa WEF system. Our goal is to understand the big picture of Iowa's WEF system by evaluating the structure of the model synthetically across scales.

The network structures could be interpreted through the arcs between nodes, which represented the corresponding connections between variables. The nodes were color-grouped by the hierarchical clustering algorithm based on node-wise modularity calculated from degree of nodes to represent community structures (Clauset et al., 2004). The models at higher resolution (top left) tended to focus on the connection among water and weather, while the models at lower resolution (bottom right) tended to focus on the connections between the biophysical system and the human system as more human-relative data become available. By considering both leading and lagging effects within and between the biophysical system and the human system, the connections at higher resolutions were assumed to be more reliable than those at lower resolutions. Based on the analysis of the feature dimension and the sampling density, we analyzed the models at different spatial scales by each of the same temporal scales to understand the transformation in the order of decreasing resolution.

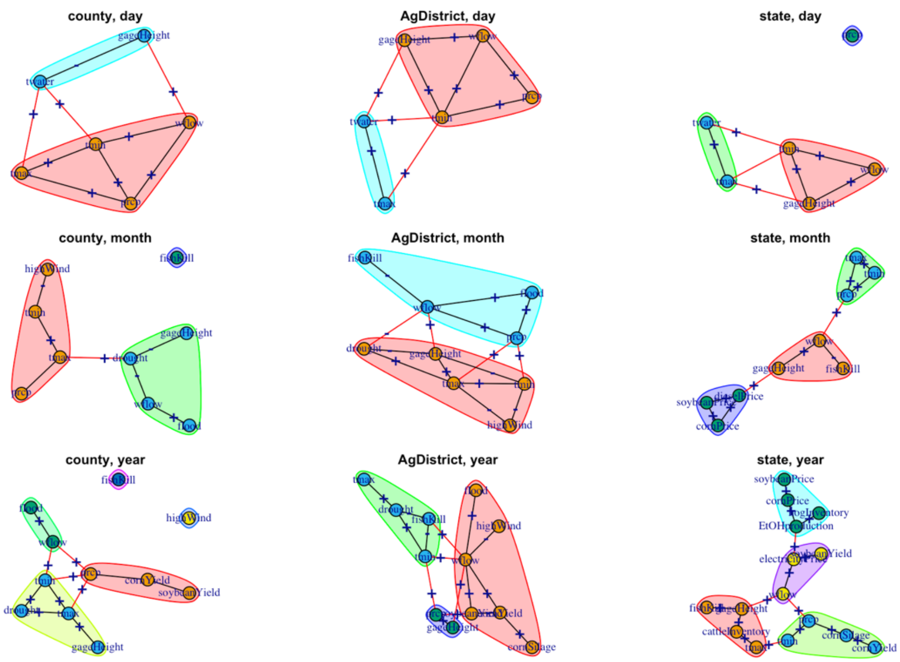

Figure 4: The result of sparse Markov network at nine levels of spatiotemporal resolutions with communities groups colored by modularity-based hierarchical clustering with "+" for positive and "-" for negative Pearson correlation for each edge. The width of edges reflects the strength of inferred relationships. The three state-scale models only describe the dependencies between variables at certain temporal scales, because only Iowa is considered.

Our model provided insight for the scale-related connections. At the top row in Figure 4, the three daily-scale models demonstrated the connections inferred from the highest resolution variables. Some connections were consistent across daily scale, which represented the instantaneous effect of the hydrologic activities between water conditions and ambient conditions. For example, the water temperature always connected with the ambient conditions including maximum and minimum air temperature, and gage height always connected with water flow rate and ambient conditions. In contract, state-scale masked the connection from precipitation to water flow rate and ambient conditions, which suggested that precipitation is less related to these factors. In fact, precipitation decreased from east to west across the state, so this partition of precipitation from the main network implied space-related lurking factors. Moreover, the daily-scale models were limited to the six hydrologic responses, which reflect on the fact that the Iowa's hydrologic data including water condition and ambient conditions had a great availability. As more data become available at lower resolution, the models were able to illustrate the relationship between the biophysical system and the human system.

While the hydrologic effect preserved, newly added variable brought more detail to the network in two ways. One way is to explain potential causation of the four environmental events, including drought, flood, high wind, and fishkill. Drought events were caused by maximum temperature, the gage height and water flow rate together, demonstrated at county- monthly and AgDistrict-monthly scales. Flood event was always caused by water flowrate at county-monthly scale, and by overabundant precipitation at AgDistrict-monthly scales. High wind event often occurs when ambient temperature was lower, which reflected on the development of prominent high wind events in association with extratropical cyclones in Midwest area. Moreover, the physical connection between fishkill event and water flowrate could not be intuitively presented. It could be justified by adding variables related to leakage from animal farms or fertilized lands, because about 48% of the fish kill events in Iowa were caused by anthropogenic activities (Iowa DNR Fish Kill Database, n.d.)

The other way is to form clique relations by clustering variables to indicate weaker connections. The models in both monthly and yearly scale had branch structures, which were colored with clusters. To understand the interlinkage in Iowa's WEF system, the variables were grouped in five categories including weather, water, food, energy, event and economic, with detail listed in APPENDIX A. More obvious the lower resolution models, especially at AgDistrict and state scales, the variable in a same cluster are most likely from the same resource sectors. Noticeably, event node were scattered between the hydrologic node at the monthly-scale. The other variables for food, energy and economic are on the branch extended from the hydrologic cluster. To verify the role hydrologic variables of the WEF system, the next section will have analysis and discussion using network analysis.

3.6.3 Network Analysis

After the network representation, we preform network analysis on the models to provide a standard interpretation of the connections. We evaluated the model using three features of network analysis, including degree distribution, assortativity analysis, and betweenness centrality. Our goal was to characterize the network structure of Iowa's WEF system.

Degree of a node measured the total number of connections to the particular node in a network. When the degrees distribution of a network followed a power law, the network was called scale-free network. The scale-free characteristics emphasized that the network consisted of a few nodes that were highly connected by other nodes. The failures of the highly connected nodes could have a wider impact on the bigger system. In our example, the degree of these network was range from 0 to 4, with the maximum count appear at the state-yearly model., none of the models shown scale-free feature, shown in Figure 5. They did not appear to follow power law, which suggest that system could have good resiliency. However, an in-depth analysis had to be conducted, because the models from 1000 bootstraps could be accurate but no precise enough.

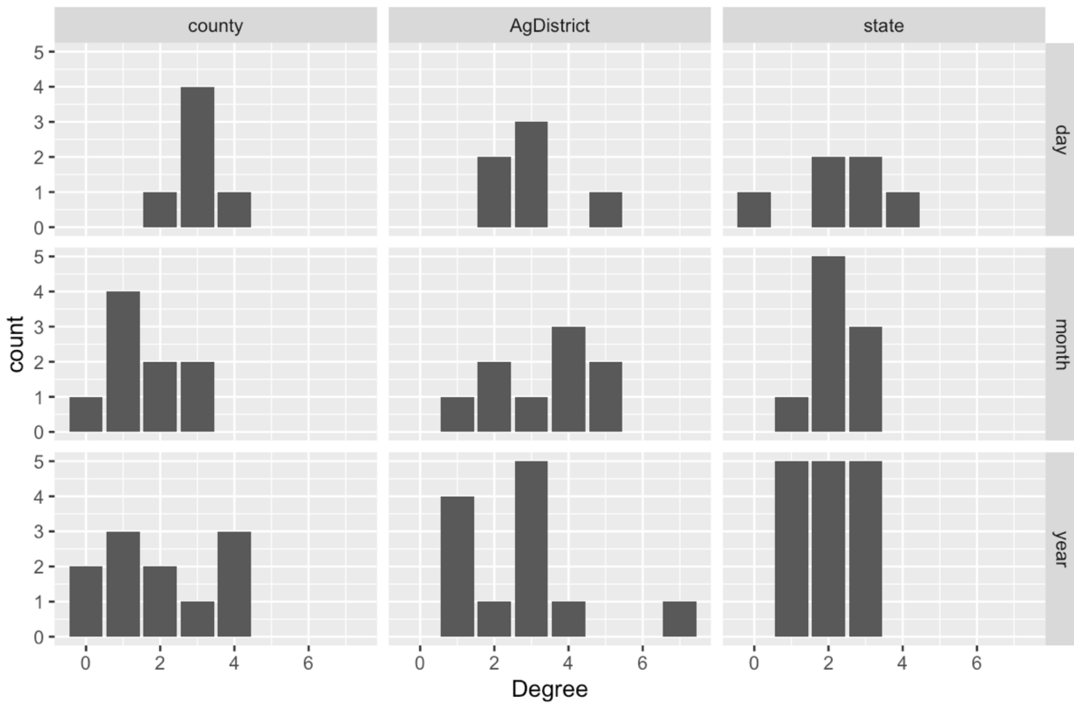

Figure 5: The histogram of degree distribution at nine spatiotemporal resolution.

The assortativity analysis of the models can provide further structural information about the system to determine if the “hubs” components existed. A network is disassortative when high degree nodes tend to attach to low degree nodes. The assortativity coefficient is a measure of the level of homophily of the graph using the Pearson correlation coefficient of degree between pairs of linked nodes in a network structure (Newman, 2003). As the result, the assortativity coefficients were below zero, indicating the network is disassortative, shown in Figure 6. The disassortative networks match the understanding of the WEF nexus perspective that each of the resource sectors tangles with each other. The disassortative of the system suggest that the global “hubs” components might not exist. Without prominent evidence of the existence of “hub” variables from the general network analysis, we moved on to understand the role of each category of variables.

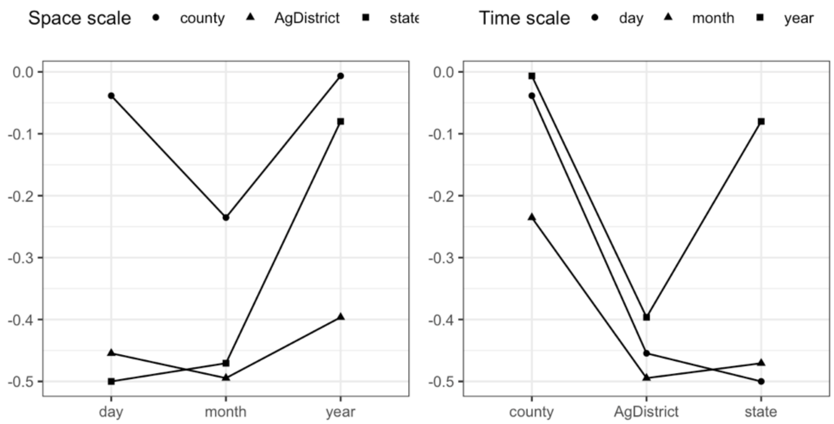

Figure 6: the comparison of the networks on the assortativity coefficient between temporal scale (left) and spatial scale (right).

Although the global “hubs” components might not exist, instead, we evaluated the existence of the local “hubs” components. The centrality measurement of the network analysis provided functionality to make comparisons between categories of grouped variables. We used betweenness centrality to describe the strength that the group of vertices are between other groups. Show in Figure 7, The higher average betweenness of the variable group of water and weather suggested that the hydrologic system was more central in the network. In comparison, the low average betweenness of the group of food and event show that these resource systems are more likely at the edge of the network. The hydrologic system being centric suggested that the change in the other sub systems would spread to hydrologic system.

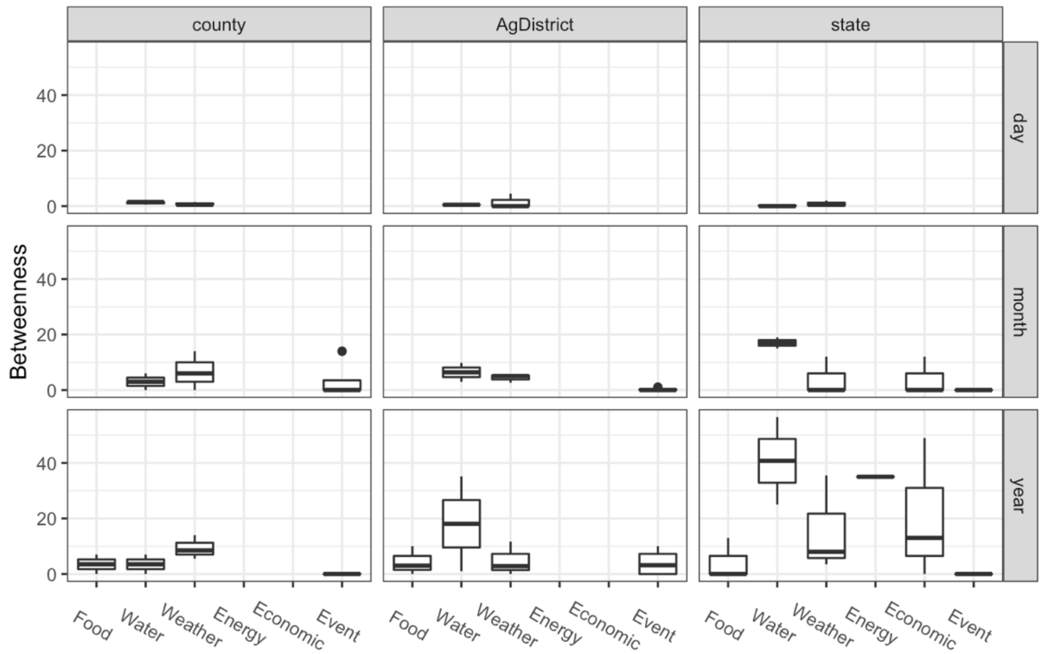

Figure 7: the average betweenness of variables categorized in Food, hydrology (water, weather), energy, economy and event.

3.6.4 Water-Centric vs. Multi-Centric

The above network model and analysis supported on the need of water-centric management for Iowa's WEF resource system. Contrasting to the multi-centric concept of nexus, water-centrism was a management perspective that emphasized the importance of water over other resources. The water-centrism was mentioned in many previous studies. Most of these studies focused on the area where the agriculture portion was heavy in the local economy. Similarly, our study focused on the agricultural-intensive area. Our result supported the reciprocal relation between agricultural and water resource systems. The network model showed that farm production was relative to the ambient condition, the water quantity, and the water quality, indicated by precipitation, flowrate, and environmental event, respectively. Moreover, the large-scale hydrologic condition subtly affected agricultural output and extends to biorientable energy production; the farm-scaled agricultural practice also directly affects the local water quality. Interacting with other resource sectors across scales, the hydrologic system can characterize the status of various natural resources in a WEF system.

However, the role of hydrologic system might not be so significant in urban area as in rural area, due to the complexity of the WEF system. The complexity reflected on both the quality and quantity of mass transfer between resource systems. The dominating mass transfers in urban areas could happen via human-made infrastructures, such as public transportation. They were so highly standardized to be efficient, so their roles could be comparable or even exceed the role of the hydrologic system. Hence, many urban nexus studies included extra elements to provide a holistic understanding of the urban resource system. Using the generic tool, such as the input-output model and life cycle assessment, could simulate some scenarios. Still, the real-time dynamic had to come from the data, such as logistic and transportation. However, the issues related to data availability and heterogeneity were more challenges for urban areas than rural areas. In this study, we demonstrated a framework consisting of data integration, modeling, and analysis. This method can help policymakers understand the interlinkage between the resource system at different spatiotemporal resolutions to customize the policy related to resource management.

3.6.5 Limitation

3.6.5.1 Data Availability

Similar to all data-driven methods, the performance of this modeling technique limited by data availability. Data availability influenced the model in two ways. One way was variable selection. As mentioned in the method section, a fixed criterion was used in the variable selection procedure. If the tested variables significantly reduced the remaining data points, it was abandoned. There was no consistent method of variable selection which included all collected data into the analysis. As a result, the hydrologic data as point data dominated the analysis at the high resolutions. The interlinkages between the biophysical system and the human system would not appear at the higher resolution until economic data become greater available.

Another way was lurking variables. Lurking variables are unmeasured variables that are responsible for an apparent correlation between two other variables. In this study, spatial pattern and temporal pattern probably the most needed lurking variables. An example of spatial patterns is precipitation, which decreases from east to west across the state. An example of temporal patterns is corn yield, which has increased since 1980. As the consequence of lurking variables, networks can separate into two parts. A better practice is to create extra variables, such as longitude, latitude and timestamp, to concern about the effect of time and space individually on the WEF nexus.

3.6.5.2 Uniparameter Assumption

The data with multiparameter distributions might violate the uniparameter assumption, which could lead to misrepresentation of the system. The model fitted a local-Poisson distribution at each node, which assumed the all node-conditional distributions were uniparameter. Uniparameter distributions, such as Poisson distribution, had assumptions on equal mean and variable for positively skewed count data; where two-parameter distribution, such as Log-normal distribution or gamma distribution, can finely control the location and scale for continuous positively skewed variables. Choosing the uniparameter assumption was a compromised decision between computation power and accuracy. The variable in this dataset cover both continuous and count variables which followed a wide range of distribution, including overdispersed, skewed characteristics. Using identical distribution for heterogeneous variables in the probabilistic graphical model was always a concern because a situation that all variables have identical distribution was almost impossible. Further, Processing gigabit of high dimension data for bootstrapping was computation expensive. To lower the computation expense, discretizing variables help to reduce the mismatch between continuous variables and uniparameter assumption of Poisson distribution.

3.6.5.3 Multiscale verses. Cross-scale

Multiscale analysis could illustrate the connection among variables, space and time, but it might not well demonstrate the connections between variables across different resolutions. For example, the weather during planting season could critically affect the corn yield, and especially the silking stage relies on the mid-May temperature (Westcott & Jewison, 2013). In this multiscale analysis, the aggregation of monthly precipitation through upscaling weaken the correlation to the corn yield appear in the yearly scale. Consequently, the analysis could not point out the relation between weather at specific month period and yearly corn yield. The analysis at cross-scale instead of multiscale was required to further reveal these connections across different temporal scales.

3.7 Conclusion

The goal of this study was to propose a framework that modeled and analyzed a WEF system and visualized the interlinkages of the nexus to support interpretation and communication. This study summarized the data availability at different scales where hydrologic data was recorded at the highest resolution. The multiscale analysis provided structural information to reveal the complexity of WEF system. For example, hydrologic system is likely to reflect the change in other sub systems in Iowa’s WEF system. More importantly, the discipline of nexus study was so broad that the research required experts from many backgrounds. The intuitive visualizations provided by the sparse network model helped to close the gap of communications. The ultimate goal of the nexus approach was to guide resource management decisions to reach systemic resilience and synergy. Our study identified the uniqueness of the hydrologic system in Iowa's WEF system. Making decision associated with the importance components in a WEF system could help to achieve these goals.

3.8 References

Clauset, A., Newman, M. E. J., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, *70*(6), 066111. https://doi.org/10.1103/PhysRevE.70.066111

Daher, B. T., & Mohtar, R. H. (2015). Water–energy–food (WEF) Nexus Tool 2.0: Guiding integrative resource planning and decision-making. *Water International*, *40*(5–6), 748–771. https://doi.org/10.1080/02508060.2015.1074148

Endo, A., Yamada, M., Miyashita, Y., Sugimoto, R., Ishii, A., Nishijima, J., Fujii, M., Kato, T., Hamamoto, H., Kimura, M., Kumazawa, T., & Qi, J. (2020). Dynamics of water–energy–food nexus methodology, methods, and tools. *Current Opinion in Environmental Science & Health*, *13*, 46–60. https://doi.org/10.1016/j.coesh.2019.10.004

Iowa Code §§ 352.1 to 352.13 (2019). https://www.legis.iowa.gov/publications/search/document?fq=id:1093257&pdid=1075323&q=352#

Iowa Corn Growers Association. (2016). *Corn Facts*. Iowa Corn. Retrieved June 15, 2020, from https://www.iowacorn.org/media-page/corn-facts

Iowa DNR Fish Kill Database. (n.d.). *Fish Kill Event Summaries*. Retrieved July 31, 2020, from https://programs.iowadnr.gov/fishkill/Summaries

Iowa Pork Producers Association. (2019). *Iowa Pork Facts*. Iowa Pork. Retrieved April 11, 2020, from https://www.iowapork.org/news-from-the-iowa-pork-producers-association/iowa-pork-facts/

Kling, C. L., Arritt, R. W., Calhoun, G., & Keiser, D. A. (2017). Integrated Assessment Models of the Food, Energy, and Water Nexus: A Review and an Outline of Research Needs. *Annual Review of Resource Economics*, *9*(1), 143–163. https://doi.org/10.1146/annurev-resource-100516-033533

Khan, Z., Linares, P., Rutten, M., Parkinson, S., Johnson, N., & García-González, J. (2018). Spatial and temporal synchronization of water and energy systems: Towards a single integrated optimization model for long-term resource planning. *Applied Energy*, *210*, 499–517. https://doi.org/10.1016/j.apenergy.2017.05.003

KTH (2013). CLEWs – climate, land, energy and water strategies to navigate the nexus. Last accessed on July 2, 2014. Retrieved from http://www.kth.se/en/itm/inst/energiteknik/forskning/desa/researchareas/clews-climate-land-energy-and-water-strategies-to-navigate-the-nexus-1.432255

National Agriculture in the Classroom. (2020, June). *A Look at Iowa Agriculture*. https://www.agclassroom.org/teacher/stats/iowa.pdf

Newman, M. E. J. (2003). Mixing patterns in networks. *Physical Review E*, *67*(2), 026126. https://doi.org/10.1103/PhysRevE.67.026126

USDA National Agricultural Statistics Service. NASS - Quick Stats. USDA National Agricultural Statistics Service. https://data.nal.usda.gov/dataset/nass-quick-stats. Accessed 2020-05-31.

Wan, Y.-W., Allen, G. I., Baker, Y., Yang, E., Ravikumar, P., Anderson, M., & Liu, Z. (2016). XMRF: An R package to fit Markov Networks to high-throughput genetics data. *BMC Systems Biology*, *10*(3), 69. https://doi.org/10.1186/s12918-016-0313-0

Westcott, P. C., & Jewison, M. (2013). Weather Effects on Expected Corn and Soybean Yields. *Economic Research Service/USDA*, *No. FDS-13G-01*, 33. https://www.ers.usda.gov/publications/pub-details/?pubid=36652

Yang, E., Allen, G., Liu, Z., & Ravikumar, P. K. (2012). Graphical Models via Generalized Linear Models. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1358–1366). Curran Associates, Inc. http://papers.nips.cc/paper/4617-graphical-models-via-generalized-linear-models.pdf

Zhang, X., & Vesselinov, V. V. (2017). Integrated modeling approach for optimal management of water, energy and food security nexus. *Advances in Water Resources*, *101*, 1–10. https://doi.org/10.1016/j.advwatres.2016.12.017

Menne, Matthew J., Imke Durre, Bryant Korzeniewski, Shelley McNeal, Kristy Thomas, Xungang Yin, Steven Anthony, Ron Ray, Russell S. Vose, Byron E.Gleason, and Tamara G. Houston (2012): Global Historical Climatology Network - Daily (GHCN-Daily), Version 3. PRCP, TMAX, TMIN. NOAA National Climatic Data Center. doi:10.7289/V5D21VHZ

USDA National Agricultural Statistics Service. (2017). NASS - Quick Stats. USDA National Agricultural Statistics Service

Read, E. K., Carr, L., De Cicco, L., Dugan, H. A., Hanson, P. C., Hart, J. A., … Winslow, L. A. (2017). Water quality data for national-scale aquatic research: The water quality portal. *Water Resources Research*, **53**, 1735– 1745. https://doi.org/10.1002/2016WR019993

Iowa Dept. of Natural Resources, 2013, Fish Kill Database. https://programs.iowadnr.gov/fishkill

STATISTICS. (n.d.). Iowa Renewable Fuels Association. Retrieved June 19, 2020, from https://iowarfa.org/resource-center/statistics/

Murphy, J. D. (Mar 9, 2016). *Storm Data Preparation* (National Weather Service Instruction 10-1605). Retrieved on June 24, 2020 from http://www.nws.noaa.gov/directives/sym/pd01016005curr.pdf

Iowa Department of Natural Resources, Environmental Services Division, Water Quality Bureau, Water Quality Monitoring & Assessment Section. (2020, February). *Methodology for Iowa's 2018 Water Quality Assessment, Listing, and Reporting Pursuant to Sections 305(b) and 303(d) of the Federal Clean Water Act.* State Library of Iowa. http://publications.iowa.gov/31281/1/2018%20IA%20Methodology-Final.pdf

Appendix: Data Description and Sources With Details Of Spatiotemporal Upscaling

Group	Variable	Description	Unit	Space Scale	Time Scale	Time Range	Spatial downscaling	Temporal downscaling	Source
Weather	prcp	Total precipitation recorded at weather station	millimeter	point	day	1980-01-01 to 2019-11-16	mean	sum	NOAA Global Historical Climatology Network
	tmax	Maximum temperature recorded at weather station	degree kelvin	point	day	1980-01-01 to 2019-11-16	max	max	NOAA Global Historical Climatology Network
	tmin	Minimum temperature recorded at weather station	degree kelvin	point	day	1980-01-01 to 2019-11-16	min	min	NOAA Global Historical Climatology Network
Food	cornYield	Corn yield	bushels per acre	county	year	1980 to 2019	mean	NA	USDA National Agricultural Statistics Service
	soybeanYield	Soybean yield	bushels per acre	county	year	1980 to 2019	mean	NA	USDA National Agricultural Statistics Service
	cattleInventory	Cattle Inventory report at the frist of January at each year	count of head	state	year	1968 to 2019	NA	NA	USDA National Agricultural Statistics Service
	hogInventory	Total hog inventory reported at the frist of January at each year	count of head	state	year	1968 to 2019	NA	NA	USDA National Agricultural Statistics Service
	eggProduction	Total egg production	count of egg	state	month	2007-12 to 2020-04	NA	sum	USDA National Agricultural Statistics Service
	cornSilage	Total silage production	tons per acre	AgDistrict	year	1980 to 2019	mean	NA	USDA National Agricultural Statistics Service
Water	wflow	Mean Stream flow collected at water sampling site	cubic feet per second	point	day	1980-01-01 to 2019-12-31	mean	mean	Water Quality Portal
	gageHeight	Mean gage height collected at water sampling site	feet	point	day	1980-01-01 to 2019-12-31	mean	mean	Water Quality Portal
	twater	Mean Water temperature collected at water sampling site	degree kelvin	point	day	1980-01-01 to 2019-12-31	mean	mean	Water Quality Portal
	Ecoli	The concentration of Escherichia Coli in water	MPN/100ml	point	day	1999-06-01 to 2019-08-28	mean	mean	Iowa DNR, AQuIa
	dissolvedOxygen	The concentration of oxygen dissolved in water	mg/l	point	day	2000-05-22 to 2013-08-28	mean	mean	Iowa DNR, AQuIa
	microcystin	The concentration of Microcystin in water	ug/l	point	day	2006-05-22 to 2019-08-28	mean	mean	Iowa DNR, AQuIa
	turbidity	The cloudiness or haziness due to the presence of suspended particulates	NTU	point	day	2000-05-22 to 2013-08-28	mean	mean	Iowa DNR, AQuIa
Energy	EtOHproduction	Iowa Ethanol Production	millions of gallons	state	year	1980 to 2019	NA	NA	Iowa Renewable Fuels Association
	biodieselProduction	Iowa Biodiesel Production	millions of gallons	state	year	2005 to 2019	NA	NA	Iowa Renewable Fuels Association
	electricGeneration	Net Generation by all type of souce including Petroleum, Hydroelectric Conventional, Wind, Coal, Natural Gas, Other Biomass.	Megawatt Hours	state	month	2001-01 to 2012-01	NA	sum	EIA State-level generation and fuel consumption data (EIA-923)
	electricitySale	Total electricity sale including Residential, Commercial, Industrial, Transportation, and Other.	Megawatt Hours	state	month	2001-01 to 2020-02	NA	sum	EIA Monthly Electric Power Industry Report (EIA-861M, formerly EIA-826)
Economic	govRevenue	State of Iowa total revenue	dollar	state	month	2012-01 to 2020-12	NA	sum	State of Iowa's data portal
	govExpenditure	State of Iowa total expenditure	dollar	state	month	2004-01 to 2020-12	NA	sum	State of Iowa's data portal
	employment	Total employment in Iowa including agriculture sector	count of employed worker	county	month	2011-01 to 2019-12	sum	mean	IWD Quarterly Census of Employment and Wages
	cornPrice	Cash corn prices	dollar	state	month	1925-01 to 2018-12	NA	mean	USDA NASS, ISU Extension and Outreach, Ag Decision Maker
	soybeanPrice	Cash soybean prices	dollar	state	month	1925-01 to 2018-12	NA	mean	USDA NASS, ISU Extension and Outreach, Ag Decision Maker
	electricityPrice	Iowa weight-average electricity price of residential, commerical, and industrial	cents/kilowatthour	state	year	1970 to 2018	NA	NA	EIA Monthly Electric Power Industry Report (EIA-861M, formerly EIA-826)
	dieselPrice	Monthly Midwest No 2 Diesel Retail Prices	Dollars per Gallon	state	month	1994-04 to 2020-05	NA	mean	EIA Gasoline and Diesel Fuel Update
Event	flood	the count of flood event	count of event	point	month	1996-01 to 2019-10	sum	sum	NOAA Storm Events Database
	drought	the count of drougth event	count of event	point	month	1996-01 to 2019-10	sum	sum	NOAA Storm Events Database
	highWind	the count of high wind event	count of event	point	month	1996-01 to 2019-10	sum	sum	NOAA Storm Events Database
	fishKill	The count of fish kill event	count of event	point	month	1981-02 to 2020-04	sum	sum	Iowa DNR Fish Kill Database

CHAPTER 4. GENERAL CONCLUSIONS

The overall objective of this research was to close the gap between nexus modeling and communication by proposing a framework of modeling and analyzing heterogeneous data in the WEF nexus discipline. Our proposed framework consists of three parts, including a data aggregation pipeline, modeling with the probabilistic graphical model (PGM), and analysis technique for network model.

The objective of chapter two was to determine the technique to model and analyze Iowa's WEF resource system. This chapter recognized the three key elements in data preparation, including data resolution, the designation of scales, upscaling arithmetic operations. According to the resolution and the physical attribute of the commonly available data, we were able to identify the candidates of spatiotemporal scales and the upscaling arithmetic operations for each variable in the data aggregation process. Importantly, this chapter identified PGM as the appropriate model in our framework that can overcome the issues in the existing methods. The comparison showed that PGM was suitable for this multiscale task in terms of methodology, the capability of scalability, and ease of communication. The network analysis was also highlighted as the primary tool to analyze the network structure of the PGM result. This chapter demonstrated the framework of data aggregation, nexus modeling, and network analysis for heterogeneous data relative to the WEF resource system.

The primary objective for chapter three was to show the capabilities of the proposed framework in identifying and representing the interlinkages and structural patterns of Iowa's WEF resource system at multiple spatiotemporal scales. With the limited data, the model was able to provide an intuitive representation to identify the significant interlinkages in Iowa's WEF resource system. The microscopic models were able to provide the highest resolution connections using the most available data. The macroscopic models were able to illustrate the relationship between the biophysical system and the human system. The network analysis of the model was able to successfully characterize the network at the variable level and the structure level. The centrality analysis and the assortativity analysis were able to make comparisons between categories of grouped variables and provide structural information about the system, respectively. As a result, Iowa's WEF resource system was characterized as a not-scale-free, disassortative network. The characterization showed that hydrologic responses played a important role in the system. The study also led to support water-centrism for Iowa's resource management, which contradicted with the current nexus philosophies of resource management. Further limitations of the framework were discussed, including data availability, uniparameter assumption in the sparse model, and multiscale paradigm.

Overall, the research from the two chapters elaborated on the framework in detail and demonstrated an application to provide a better understanding of its capabilities. With Iowa's example, we provided an overview of data availability for the rural WEF system and demonstrated a feasible solution to integrate, analyze, and visualize heterogeneous data using our framework. This framework is useful to close the gap between nexus modeling and communication by fundamental knowledge of statistics and intuitive representation of the system.

APPENDIX A: NECESSARY FUNCTIONS TO RUN ANALYSIS

```
# vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  query water
          vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv


# query site information for state around IA
read_siteMidWest_info <- function(states, update){
  if (update ==TRUE){
    siteMidWest_info <- plyr::mdply(states, function(x) {
      site_info_new  <- readNWISdata(stateCd = x, service="site", seriesCatalogOutput=TRUE,
                                startDate = "2000-01-01",
                                endDate = "2018-12-31")}) %>%
      select(-X1) %>%
      as_tibble() %>%
      mutate_at(.vars = c("begin_date", "end_date"), ymd)
    saveRDS(siteMidWest_info, file = "data/siteMidWest_info.Rds")
  } else if (update== FALSE){
    siteMidWest_info <- readRDS("data/siteMidWest_info.Rds")
  }
  return(siteMidWest_info)
}

# querydatas function
querydatas <- function(site = site,
                       pCode = pCode,
                       startDate = startDate,
                       endDate = endDate){

  querydata <- function(site){

    if (site$data_type_cd =="dv"){
      tmp_1 <- try(readNWISdv(siteNumbers = site$site_no,
                             parameterCd = site$parm_cd,
                             startDate = site$startDate,
                             endDate = site$endDate),
                  silent = FALSE)
      if (nrow(tmp_1) == 0) return(NULL)
      tmp_2 <- tmp_1 %>%
        as_tibble %>%
        select(Date, paste("X", site$parm_cd, site$stat_cd, sep = "_")) %>%
        setNames(c("sample_dt", "result")) %>%
        mutate(sample_dt = lubridate::as_date(sample_dt))
```

```
    } else if (site$data_type_cd =="qw"){
      tmp_1 <- try(readNWISqw(siteNumbers = site$site_no,
                              parameterCd = site$parm_cd,
                              startDate = site$startDate,
                              endDate = site$endDate,
                              expand = FALSE),
                   silent = FALSE)
      if (nrow(tmp_1) == 0) return(NULL)
      tmp_2 <- tmp_1 %>%
        as_tibble %>%
        select(sample_dt, paste("p", site$parm_cd, sep = "")) %>%
        setNames(c("sample_dt", "result")) %>%
        mutate(sample_dt = lubridate::as_date(sample_dt))
    }

    return(tmp_2)
  }

  plan(multiprocess) # switch to parallel computing
  parm_data <- siteMidWest_select %>%
    filter(parm_cd == pCode) %>%
    tibble::rowid_to_column("index") %>%
    select(index, site_no, parm_cd, data_type_cd, stat_cd) %>%
    mutate(startDate = startDate) %>%
    mutate(endDate = endDate) %>%
    nest(site = c(site_no, parm_cd, data_type_cd, stat_cd, startDate, endDate)) %>%
    mutate(data = site %>% furrr::future_map(possibly(querydata, NA_real_))) %>%
    unnest(site) %>%
    select(-index); parm_data

  plan(sequential) # back to sequential computing

  return(parm_data)
}
# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  query water   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

# vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  Function   vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv

thinshp <- function(shp){
  shp_st <- maptools::thinnedSpatialPoly(
    as(shp, "Spatial"), tolerance = 0.1,
    minarea = 0.001, topologyPreserve = TRUE)
  shp <- st_as_sf(shp_st)
  return(shp)
}
```

```
site2sf <- function(df,id_cn ="id", Lon_cn = "longitude", Lat_cn = "latitude", crs= 4326){
  # cleanup data
  df_locs <- df %>% rename(longitude=Lon_cn, latitude=Lat_cn, id = id_cn)
  # convert to sf object
  df_locs <- st_as_sf(df_locs,
                      coords = c("longitude", "latitude"), # for point data
                      remove = F, # don't remove these lat/lon cols from df
                      crs = crs) # add projection (this is WGS84)
  return(df_locs)
}


# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  Function   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

# vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv point data preprocessing
        vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
crop_data_by_boundary <- function(data_sf, boundary = "ia_sf" ){

  site2sf_nest <- function(df, id_cn ="id", Lon_cn = "longitude", Lat_cn = "latitude", crs= 4326){
    #nest the unrenamed colummns
    df_locs <- df %>%
      rename(id = id_cn, longitude = Lon_cn, latitude = Lat_cn) %>%
      nest(data = c(-id, -longitude, -latitude))
    # convert to sf object
    df_locs <- st_as_sf(df_locs,
                        coords = c("longitude", "latitude"), # for point data
                        remove = F, # don't remove these lat/lon cols from df
                        crs = crs) # add projection (this is WGS84)
    return(df_locs)
  }

  sf2site_unnest <- function(df){
    df %>%
      sf::st_drop_geometry() %>%
      select(-region) %>%
      unnest(cols = c(data))
  }

  data_sf <- data_sf %>%
    site2sf_nest() %>%
    st_intersection(get(boundary, envir = .GlobalEnv)) %>%
    sf2site_unnest()
  return(data_sf)
}

crop_region_by_boundary <- function(region_sf, boundary = "ia_sf"){
```

```
  boundary <- get(boundary, envir = .GlobalEnv) %>% st_geometry()
  region_sf <- get(region_sf, envir = .GlobalEnv) %>% sf::st_intersection(boundary)
  return(region_sf)
}

rescale_point_data <- function(x, to = c(1, 5)){
  x %>%
    group_by(id) %>%
    mutate_at(names(x)[5] , scales::rescale, to = to) %>%
    ungroup()
}
# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ point data preprocessing ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

# vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv downscaling Function
        vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv

## spatiotemporal rescaling = function(data_sf, time_rescale, space_rescale, sf_geometry)
point_downscaling <- function(point_data, to_region = NULL, to_period = NULL,
        Spatial_downscaling = "mean", Temporal_downscaling = "mean"){
  # test: data <- prcp; sf_geometry <- county_sf; Spatial_downscaling = "mean";
        Temporal_downscaling = "mean"
  # test: to_region = "county_sf"; to_period = "day"
  assign_region <- function(data, sf_geometry){
    site2sf_nest <- function(df, id_cn ="id", Lon_cn = "longitude", Lat_cn = "latitude", crs=
        4326){

      # # nest the unrenamed colummns
      df_locs <- df %>%
        rename(id = id_cn, longitude = Lon_cn, latitude = Lat_cn) %>%
        nest(data = c(-id, -longitude, -latitude))
      # convert to sf object
      df_locs <- st_as_sf(df_locs,
                          coords = c("longitude", "latitude"), # for point data
                          remove = F, # don't remove these lat/lon cols from df
                          crs = crs) # add projection (this is WGS84)
      return(df_locs)
    }
    suppressWarnings({
      # convert df to sf object # intersect the site and polygon
      data_sf <- data %>%
        site2sf_nest() %>%
        st_intersection(x = ., y = sf_geometry) %>%
        st_drop_geometry() %>%
        unnest(data) %>%
        select(-id, -longitude, -latitude)
    })
```

```
    return(data_sf)
  }
  convert_to_period <- function(df, to_period){
    if (to_period == "day"){
      return(df)
    } else if (to_period == "month"){
      df <- df %>%
        mutate(Date = Date %>% as.character() %>%
                 stringr::str_sub(end = 7) %>%
                 lubridate::ymd(truncated = 1))
    } else if(to_period == "year"){
      df <- df %>%
        mutate(Date = Date %>% as.character() %>%
                 stringr::str_sub(end = 4) %>%
                 lubridate::ymd(truncated = 2))
    }
    return(df)
  }
  result_data <- point_data %>%
    get(envir = .GlobalEnv) %>%
    assign_region(sf_geometry = to_region %>% get(envir = .GlobalEnv)) %>%
    group_by(Date, region) %>%
    summarise_all(Spatial_downscaling, na.rm = TRUE) %>% # Spatial_downscaling
    ungroup()
  if (to_period != "day"){
    result_data <- result_data %>%
      convert_to_period(df = ., to_period = to_period) %>%
      group_by(region, Date) %>%
      summarise_all(Temporal_downscaling, na.rm = TRUE) %>%
      ungroup()
  }
  result_data <- result_data %>%
    dplyr::rename(period = "Date") %>%
    mutate(period = period %>% as.character %>% as.factor) %>%
    mutate(region = region %>% as.factor) %>%
    as.data.frame()
  return(result_data)
}

# downscaling initial dataset by pointDataSummary
initial_pointData_downscaling <- function(point_datas = NULL, periods = periods, regions =
        regions, Spatial_downscalings = NULL, Temporal_downscalings = NULL){
  # test: period <- periods[1]; region <- regions[1]
  for (period in periods){
    for (region in regions){
      cat("\n\n")
```

```
    scale_label <- paste0(stringr::str_sub(region, end = -4L), "X", period)
    print(paste0("pointdata_", scale_label))
    tibble(point_data = point_datas,
           Spatial_downscaling = Spatial_downscalings,
           Temporal_downscaling = Temporal_downscalings) %>%
      mutate(to_period = period, to_region = region) %>%
      select(point_data, to_region, to_period, Spatial_downscaling,
         Temporal_downscaling) %>%
      purrr::pmap(point_downscaling) %>%
      purrr::reduce(full_join, by = c("region", "period")) %>%
      mutate(period = as.factor(period)) %>%
      mutate(region = as.factor(region)) %>%
      select(region, period, everything()) %>%
      arrange(region, period) %>%
      assign(value = .,
             x =  paste0("pointdata_", scale_label),
             envir = .GlobalEnv)
    cat("dim: ");
    get(paste0("pointdata_", scale_label), envir = .GlobalEnv) %>% dim %>% cat()
   }
  }
}

# downscaling multiple data by pointDataSummary
point_data_downscaling <- function(point_datas = NULL, periods = periods, regions = regions,
         Spatial_downscalings = NULL, Temporal_downscalings = NULL){
  # test: n= 1
  for (n in 1:length(point_datas)){
    point_data <- point_datas[n]
    Spatial_downscaling <- Spatial_downscalings[n]
    Temporal_downscaling <- Temporal_downscalings[n]

    # test: period <- periods[1]; region <- regions[1]
    for (period in periods){
      for (region in regions){
        cat("\n")
        scale_label <- paste0(stringr::str_sub(region, end = -4L), "X", period)
        print(paste0(point_data,"_", scale_label))
        tibble(point_data = point_datas,
               Spatial_downscaling = Spatial_downscalings,
               Temporal_downscaling = Temporal_downscalings) %>%
          mutate(to_period = period, to_region = region) %>%
          select(point_data, to_region, to_period, Spatial_downscaling,
            Temporal_downscaling) %>%
          purrr::pmap(point_downscaling) %>%
          purrr::reduce(full_join, by = c("region", "period")) %>%
```

```
        mutate(period = as.factor(period)) %>%
        mutate(region = as.factor(region)) %>%
        select(region, period, everything()) %>%
        arrange(region, period) %>%
        assign(value = .,
               x =  paste0(point_data,"_", scale_label),
               envir = .GlobalEnv)
      cat(paste("# row: ", nrow(get(paste0(point_data,"_", scale_label), envir = .GlobalEnv) )))
    }
  }
 }
}



polygon_data_downscaling <- function(polygon_data_dict = polygon_data_dict ,
        regional_division_df= NULL){
  # test: nr = 31
  convert_to_period <- function(df, period){
    if (period == "day"){
      return(df)
    } else if (period == "month"){
      df <- df %>%
        mutate(period = period %>% as.character() %>%
               stringr::str_sub(end = 7) %>%
               lubridate::ymd(truncated = 1))
    } else if(period == "year"){
      df <- df %>%
        mutate(period = period %>% as.character() %>%
               stringr::str_sub(end = 4) %>%
               lubridate::ymd(truncated = 2))
    }
    return(df)
  }
  for (nr in 1:nrow(polygon_data_dict)){
    # area_data name
    data <- polygon_data_dict[nr, ]$var
    # original scale
    orig_period <- polygon_data_dict[nr, ]$orig_period
    orig_region <- polygon_data_dict[nr, ]$orig_region
    # destinate scale
    period <- polygon_data_dict[nr, ]$dest_period
    region <- polygon_data_dict[nr, ]$dest_region
    scale_label <- paste0(region, "X", period)
    # downscaling method
```

```
Spatialdownscaling <- polygon_data_dict[nr, ]$Spatialdownscaling
Temporaldownscaling <- polygon_data_dict[nr, ]$Temporaldownscaling

cat(paste0(data, " %>% ", Temporaldownscaling, "() %>% ", Spatialdownscaling, "() to ",
        region, "X", period))
# obtain data
tmp_var <- get(data, envir = .GlobalEnv) %>%
  rename(period = orig_period) %>%
  rename(region = orig_region)

# formulate the period col
if (region != "day"){
tmp_var <- tmp_var %>%
  convert_to_period(df = ., period = period)
}
# Temporal downscaling
  tmp_var <- tmp_var %>%
    group_by(region, period) %>%
    summarise_all("mean", na.rm = TRUE) %>%
    ungroup()
# Spatial downscaling
if (orig_region != region){
  tmp_var <- regional_division_df %>%
    rename(dest_region = region) %>%
    rename(region = orig_region) %>%
    select(region, dest_region) %>%
    right_join(tmp_var, by = "region") %>%
    select(-region) %>%
    rename(region = dest_region) %>%
    group_by(region, period) %>%
    summarise_all(Spatialdownscaling, na.rm = TRUE) %>%
    ungroup() %>%
    mutate_at(c("region", "period"), as.factor)
}

# rm.na
tmp_var <- tmp_var %>%
  na.omit() %>%
  mutate_at(c("region", "period"), as.factor)

# assign
tmp_var %>%
  assign(value = .,
         x =  paste0(data, "_", scale_label),
         envir = .GlobalEnv)
cat(paste("# row: ", nrow(tmp_var))); cat("\n")
```

```
  }
}

add_time_space_varible <- function(x, scale_label){
  x <- x %>% left_join(scale_label %>%  # add coordinate
            strsplit(split='X', fixed=TRUE) %>%
            .[[1]] %>%
            .[1] %>%
            paste("_sf", sep = "") %>%
            get(envir = .GlobalEnv) %>%
            st_drop_geometry() %>%
            cbind(scale_label %>%
                    strsplit(split='X', fixed=TRUE) %>%
                    .[[1]] %>%
                    .[1] %>%
                    paste("_sf", sep = "") %>%
                    get(envir = .GlobalEnv) %>%
                    st_geometry() %>%
                    st_centroid() %>%
                    st_coordinates() %>%
                    as.data.frame() %>%
                    `colnames<-`(c("longitude", "Latitude"))) %>%
            mutate(longitude = longitude * 10 - min(longitude* 10)) %>%
            mutate(Latitude = Latitude * 10 - min(Latitude* 10)) %>%
            mutate_at(c("longitude", "Latitude"), round) %>%
            as.tibble(),
          by = "region") %>%
    mutate(time = period %>% yday()) %>%
    mutate(time = time - min(time))
  return(x)
}

# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  downscaling Function
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  variableSelection Function
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^


# variable selection modelue
forwardSelection <- function(scale_label, rrp = 0.8){
  checkRemainRow <- function(new_var, tmp_joined){
    tmp_joined %>%
      left_join(new_var, by = c("region", "period")) %>%
      na.omit() %>%
      nrow()
```

```
  }
  # obtain initial joined dataset
  tmp_joined <- ls(pattern = paste0("pointdata_", scale_label),
                   envir = .GlobalEnv) %>%
    get(envir = .GlobalEnv) %>%
    na.omit()
  # obtain nrow of initial joined dataset
  initialRow <- remainingRow <- tmp_joined %>% nrow(); initialRow
  # calculate the remainding nrow by adding one variable
  while (max(remainingRow) > initialRow*rrp){ # logical test with Remaining Row Percentage
    ## find add_var_index
    remainingRow <- ls(pattern = scale_label,
                       envir = .GlobalEnv) %>%
      .[!. %in% paste0("pointdata_", scale_label)] %>%
      .[!. %in% paste0(names(tmp_joined),"_", scale_label)] %>%
      map(get, envir = .GlobalEnv) %>%
      map(~checkRemainRow(new_var = .x, tmp_joined= tmp_joined)) %>%
      unlist(); remainingRow

    if (length(remainingRow) == 0) break

    add_var_index <- remainingRow %>%
      which(x = (. == max(.))); add_var_index
    # join add_var_index
    add_var <- ls(pattern = scale_label,
                  envir = .GlobalEnv) %>%
      .[!. %in% paste0("pointdata_", scale_label)] %>%
      .[!. %in% paste0(names(tmp_joined),"_", scale_label)] %>%
      .[c(add_var_index)]; add_var

    tmp_joined <- add_var %>%
      map(get, envir = .GlobalEnv) %>%
      purrr::reduce(full_join, by = c("period", "region")) %>%
      na.omit() %>%
      right_join(tmp_joined, by = c("period", "region")) %>%
      na.omit()
  }
  return(tmp_joined)
}



# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  variableSelection Function
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
#vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv Bayesian Network
          vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
# Bayesian Network
bnModeling <- function(tmp_joined){
  tmp_model <- tmp_joined %>%
    select(-region, -period) %>%
    as.data.frame() %>%
    boot.strength(data = ., algorithm = "hc") %>%
    averaged.network
}

plotMatrix_bn <- function(scale_label){
  cat("\n\n"); cat(paste0("joined_", scale_label))

  if (nrow(tmp_joined)!=0){
    get(paste0("model_", scale_label),
        envir = .GlobalEnv) %>%
      graphviz.plot(x = ., layout = "fdp",
                    main = str_replace(scale_label, pattern = "X", replacement = " by "))
  } else {
    plot(0,type='n',axes=FALSE, main = "NA", xlab="", ylab="")
  }
}

plotMatrix_moralbn <- function(scale_label){
  cat("\n\n"); cat(paste0("joined_", scale_label))

  if (nrow(tmp_joined)!=0){
    get(paste0("model_", scale_label),
        envir = .GlobalEnv) %>%
      bnlearn::moral() %>%
      graphviz.plot(x = ., layout = "fdp",
                    main = str_replace(scale_label, pattern = "X", replacement = " by "))
  } else {
    plot(0,type='n',axes=FALSE, main = "NA", xlab="", ylab="")
  }
}


#^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ Bayesian Network ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

#vvvvvvvvvvvvvvvvvvvvvvvv exponential family Markov Networks
          vvvvvvvvvvvvvvvvvvvvvvvvvv
xmrfModeling <- function(tmp_joined){
  # my dataset
  simDat <- tmp_joined %>%
```

```
    select_if(~n_distinct(.) > 1) %>%
    select(-region, -period) %>%
    as.matrix() %>%
    t
  p = nrow(simDat)
  n = ncol(simDat)
  # Compute the optimal lambda
  lmax = lambdaMax(t(simDat))
  lambda = 0.01* sqrt(log(p)/n) * lmax
  # Run: stability = "bootstrap",  retains network edges that are estimated in more than 95 %
          (sth=0.95) of the 50 bootstrap repetitions (N=1000)
  model_lpgm <- XMRF(simDat, method="LPGM", N=1000, lambda.path=lambda, stability =
          "bootstrap", sth = 0.95)
  # Run: stability="STAR"
  # model_lpgm <- XMRF(simDat, method="LPGM", nlams=20, stability="STAR", th=0.001)

  return(model_lpgm)
}
plotMatrix_xmrf <- function(scale_label){

  cat("\n\n"); cat(paste0("joined_", scale_label))
  tmp_joined <- get(paste0("joined_", scale_label), envir = .GlobalEnv) %>%
    na.omit() %>%
    select(-region, -period)
  cm <- cor(tmp_joined)
  tmp_joined <- tmp_joined %>%
    as.matrix() %>%
    t
  tmp_model <- get(paste0("model_", scale_label),
                envir = .GlobalEnv)
  if (nrow(tmp_joined)!=0){
    lpgm_igraph <- graph_from_adjacency_matrix(tmp_model$network[[1]], mode =
          "undirected", weighted = NULL,
                                 diag = TRUE, add.colnames = NULL, add.rownames = NA)

    allCor <- {}
    for (i in 1:nrow(tmp_joined)){

      tmp_cor <- cm[i,as.vector(lpgm_igraph[[i]][[1]])] %>% as.vector()

      if (length(tmp_cor)==0) {
        allCor <- append(allCor, NA)
      } else {
        allCor <- append(allCor, tmp_cor)
      }
    }
```

```
    #plot(lpgm_igraph, vertex.label=rownames(tmp_joined), main = str_replace(scale_label,
        pattern = "X", replacement = " by "))
    lpgm_cluster <- cluster_fast_greedy(lpgm_igraph)
    cat("# of ", scale_label, " cluster :", length(lpgm_cluster))
    plot(lpgm_cluster, lpgm_igraph,
       vertex.label=rownames(tmp_joined),
       edge.label = ifelse(allCor > 0, "+", "-"),
       edge.label.cex = 1.5,
       edge.label.font = 2,
       main = str_replace(scale_label, pattern = "X", replacement = ", "))

  } else {
    plot(0,type='n',axes=FALSE, main = "NA", xlab="", ylab="")
  }
}

#^^^^^^^^^^^^^^^^^^^^^^^^ exponential family Markov Networks ^^^^^^^^^^^^^^^^^^^^^^^^^^

#vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv Summarise Network
        vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv

# summarise a network
SummariseNetwork <- function(Scale){
  # test: Scale = "stateXyear"
  tmp_model <- ls(pattern = paste0("model_", Scale), envir = .GlobalEnv) %>%
    get(envir = .GlobalEnv)
  tmp_igraph <- graph_from_adjacency_matrix(tmp_model$network[[1]], mode = "undirected",
                                weighted = NULL, diag = TRUE,
                                add.colnames = NULL, add.rownames = NA)
  tmp_joined <- ls(pattern = paste0("joined_", Scale), envir = .GlobalEnv) %>%
    get(envir = .GlobalEnv) %>% na.omit() %>% select(-region, -period)
  tibble("# of Data point" = nrow(tmp_joined),
       "# of Variable" = vcount(tmp_igraph),
       "# of Arc" = ecount(tmp_igraph),
       "average degree" = mean(degree(tmp_igraph)),
       "average betweenness" = mean(betweenness(tmp_igraph)),
       "Assortativity Coefficient" = assortativity_degree(tmp_igraph, directed = FALSE))
}

degreeNetork <- function(Scale){
  dataGroup <- readr::read_csv("dataSummary.csv") %>%
    select(Group, Variable) %>% fill(Group)

  # test: Scale = "stateXyear"
  tmp_model <- ls(pattern = paste0("model_", Scale), envir = .GlobalEnv) %>%
    get(envir = .GlobalEnv)
```

```
  tmp_igraph <- graph_from_adjacency_matrix(tmp_model$network[[1]], mode = "undirected",
                                            weighted = NULL, diag = TRUE,
                                            add.colnames = NULL, add.rownames = NA)
  tmp_joined <- ls(pattern = paste0("joined_", Scale), envir = .GlobalEnv) %>%
    get(envir = .GlobalEnv) %>% na.omit() %>% select(-region, -period)
  tibble(Variable = names(tmp_joined),
         Degree = degree(tmp_igraph)) %>%
    left_join(dataGroup,by = "Variable") %>%
    select(Variable, Group, Degree) %>%
    mutate(Scale = Scale)
}

betweennessNetork <- function(Scale){
  dataGroup <- readr::read_csv("dataSummary.csv") %>%
    select(Group, Variable) %>% fill(Group)

  # test: Scale = "stateXyear"
  tmp_model <- ls(pattern = paste0("model_", Scale), envir = .GlobalEnv) %>%
    get(envir = .GlobalEnv)
  tmp_igraph <- graph_from_adjacency_matrix(tmp_model$network[[1]], mode = "undirected",
                                            weighted = NULL, diag = TRUE,
                                            add.colnames = NULL, add.rownames = NA)
  tmp_joined <- ls(pattern = paste0("joined_", Scale), envir = .GlobalEnv) %>%
    get(envir = .GlobalEnv) %>% na.omit() %>% select(-region, -period)
  tibble(Variable = names(tmp_joined),
         Betweenness = betweenness(tmp_igraph)) %>%
    left_join(dataGroup,by = "Variable") %>%
    select(Variable, Group, Betweenness) %>%
    mutate(Scale = Scale)
}
#^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ Summarise Network
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

#vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv combine Edge
            vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv


combineEdge <- function(by_scale = "none"){

  # summarise a network
  extractEdges <- function(Scale){
    # test: Scale = "countyXmonth"
    spaceScale <- strsplit(Scale, split='X', fixed=TRUE) %>% .[[1]] %>% .[1]
    timeScale <- strsplit(Scale, split='X', fixed=TRUE) %>% .[[1]] %>% .[2]
    tmp_model <- ls(pattern = paste0("model_", Scale), envir = .GlobalEnv) %>%
      get(envir = .GlobalEnv)
```

```
  tmp_joined <- ls(pattern = paste0("joined_", Scale), envir = .GlobalEnv) %>%
    get(envir = .GlobalEnv) %>% na.omit() %>% select(-region, -period)

  tmp_edges <- tmp_model$network[[1]]
  tmp_edges[lower.tri(tmp_edges)] <- NA
  tmp_edges %>%
    `rownames<-`(names(tmp_joined)) %>%
    `colnames<-`(names(tmp_joined)) %>%
    as.data.frame() %>%
    rownames_to_column(var = "V1") %>%
    pivot_longer(cols = -V1, names_to = "V2", values_to = "edgeExist") %>%
    filter(edgeExist == 1) %>%
    mutate(spaceScale = spaceScale,
           timeScale = timeScale) %>%
    select(spaceScale, timeScale, V1, V2, edgeExist)
}

scales <- ls(pattern = "model_", envir = .GlobalEnv) %>% str_remove("model_")
combEdges <- scales %>%
  map(extractEdges) %>%
  reduce(rbind)

if (by_scale == "spaceScale"){
  combEdges %>%
    mutate(edge = paste0(V1,"-", V2)) %>%
    mutate(spaceScale = spaceScale %>% fct_relevel("county", "AgDistrict", "state")) %>%
    ggplot() +
    geom_bar(aes(fct_infreq(edge))) +
    coord_flip() +
    facet_grid(. ~ spaceScale) +
    ylab("Count") +
    xlab("Edge")
}else if (by_scale == "timeScale"){
  combEdges %>%
    mutate(edge = paste0(V1,"-", V2)) %>%
    mutate(timeScale = timeScale %>% fct_relevel("day", "month", "year")) %>%
    ggplot() +
    geom_bar(aes(fct_infreq(edge))) +
    coord_flip() +
    facet_grid(. ~ timeScale)+
    ylab("Count") +
    xlab("Edge")
} else{
  combEdges %>%
    mutate(edge = paste0(V1,"-", V2)) %>%
    ggplot() +
```

```
    geom_bar(aes(fct_infreq(edge))) +
    ylab("Count") +
    xlab("Edge") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
  }
}
#^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ combine Edge ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

APPENDIX B: CODE FOR CLEAN POLYGON DATA

```
# set wd
setwd(dirname(rstudioapi::getSourceEditorContext()$path))
rm(list=ls())
dev.off()

library(sf)
library(dplyr)
library(ggplot2)
library(tidyr)
library(lubridate)

setwd(dirname(rstudioapi::getSourceEditorContext()$path))
setwd("data/raw excel data")

# import xlsx
file_names <- as.list(dir(pattern="*.xlsx"))
sheets_names <- lapply(file_names,readxl::excel_sheets)
# assign sheet name to df
for (i in 1:length(file_names)) {
  for (j in 1:length(sheets_names[[i]])) {
    assign(sheets_names[[i]][j], readxl::read_excel(file_names[[i]], sheet = sheets_names[[i]][j]))
  }
};
setwd(dirname(rstudioapi::getSourceEditorContext()$path))

# import csv
setwd("data/raw excel data"); dir();
temp = list.files(pattern="*.csv$")
tempname = sapply(strsplit(temp, split='.', fixed=TRUE), function(x) (x[1]))
# assign file name to df
for (i in 1:length(temp)) assign(tempname[i], readr::read_csv(temp[i]))
setwd(dirname(rstudioapi::getSourceEditorContext()$path))

# vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv region_sf
        vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
# county_sf
county <- sf::st_as_sf(maps::map("county", plot = FALSE, fill = TRUE)) %>%
  rename("geometry" = "geom")
county_sf <- subset(county, grepl("iowa,", county$ID)) %>%
  mutate(county = ID %>% stringr::str_replace("iowa,","")) %>%
  select(county, geometry) %>%
  mutate(county = tolower(county)) %>%
  mutate(county = gsub(x= county, " ","")) %>%
```

```
  setNames(c("region", "geometry"))

state_sf <- sf::st_as_sf(maps::map("state", plot = FALSE, fill = TRUE)) %>%
  rename("geometry" = "geom") %>%
  select(ID, geometry) %>%
  setNames(c("region", "geometry")) %>%
  mutate(geometry = geometry %>% st_transform(crs = 2163)) %>%
  mutate(geometry = geometry %>% lwgeom::st_snap_to_grid(size = 0.01) %>%
         lwgeom::st_make_valid()) %>%
  mutate(geometry = geometry %>% st_transform(crs = 4326)) %>%
  st_cast("MULTIPOLYGON")

ia_sf <- state_sf %>% filter(region == "iowa")

# AgDistrict_sf
AgDistrict <- cornYield %>% filter(Year ==2010) %>%
  select(County, `Ag District`) %>%
  setNames(c("county", "AgDistrict")) %>%
  distinct() %>%
  arrange(county) %>%
  mutate(county = gsub(x= county, " ","")) %>%
  mutate(county = tolower(county)) %>%
  right_join(county_sf, by =  c("county" = "region")) %>%
  mutate(geometry = geometry %>% st_transform(crs = 2163)) %>%
  mutate(geometry = geometry %>% lwgeom::st_snap_to_grid(size = 0.01) %>%
         lwgeom::st_make_valid()) %>%
  select(-county) %>%
  mutate(geometry = geometry %>% st_transform(crs = 4326))
AgDistrict_sf <- st_sf(AgDistrict,
                       geometry = st_sfc(AgDistrict$geometry,
                                         crs = 4326)) %>%
  setNames(c("region", "geometry")) %>%
  group_by(region) %>%
  summarise(geometry = st_combine(x = geometry))%>%
  st_union(by_feature = TRUE) %>%
  st_simplify(dTolerance = 1/1000)

regional_division_countyAgDistrictstate <- cornYield %>%
  filter(Year ==2010) %>%
  select(County, `Ag District`) %>%
  setNames(c("county", "AgDistrict")) %>%
  distinct() %>%
  arrange(county) %>%
  mutate(county = gsub(x= county, " ","")) %>%
  mutate(county = tolower(county)) %>%
  mutate(state = "iowa") %>%
```

```
  mutate_all(as.factor)

county_sf <- county_sf %>% st_simplify()

save(AgDistrict_sf,
     county_sf,
     ia_sf,
     state_sf,
     regional_division_countyAgDistrictstate,
     file = "data/region_sf.RData")

load("data/region_sf.RData", verbose = TRUE)

# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ region_sf
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^


# vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv polygon_data
          vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv

# agriculture corn, soyben price
cornPrice <-  cornPrice %>%
  rename(year = Year) %>%
  pivot_longer(cols = -year, names_to = "month", values_to = "cornPrice") %>%
  mutate(month = ifelse(month == "Sept", "Sep", month)) %>%
  mutate(month = month %>% match(.,month.abb)) %>%
  mutate(day = 1) %>%
  mutate(period = lubridate::as_date(paste(year, month, day,sep="-"))) %>%
  mutate(region = "iowa") %>%
  select(region, period, cornPrice) %>%
  rename(state = region, month = period)
soybeanPrice <- soybeanPrice %>%
  rename(year = Year) %>%
  pivot_longer(cols = -year, names_to = "month", values_to = "soybeanPrice") %>%
  mutate(month = ifelse(month == "Sept", "Sep", month)) %>%
  mutate(month = month %>% match(.,month.abb)) %>%
  mutate(day = 1) %>%
  mutate(period = lubridate::as_date(paste(year, month, day, sep="-"))) %>%
  mutate(region = "iowa") %>%
  select(region, period, soybeanPrice) %>%
  rename(state = region, month = period)


# yield
cornYield <- cornYield %>%
  select(County, Year, Value) %>%
  setNames(c("county","year", "cornYield")) %>%
```

```
  arrange(county) %>%
  mutate(county = tolower(county)) %>%
  mutate(county = gsub(x= county, "\"","")) %>%
  mutate(county = gsub(x= county, " ","")) %>%
  mutate(year = lubridate::ymd(year, truncated = 2L))
cornSilage <- cornSilage %>%
  select("Ag District", Year, Value) %>%
  setNames(c("AgDistrict","year", "cornSilage")) %>%
  mutate(AgDistrict = AgDistrict %>% as.factor()) %>%
  mutate(year = lubridate::ymd(year, truncated = 2L))
soybeanYield <- soybeanYield %>%
  select(County, Year, Value) %>%
  setNames(c("county","year", "soybeanYield")) %>%
  arrange(county) %>%
  mutate(county = tolower(county)) %>%
  mutate(county = gsub(x= county, "\"","")) %>%
  mutate(county = gsub(x= county, " ","")) %>%
  mutate(year = lubridate::ymd(year, truncated = 2L))


# this functionis for animal inventory
Period2Month <- function(Period){
  if (Period == "FIRST OF MAR") {
    Month = 3
  } else if (Period == "FIRST OF JUN") {
    Month = 6
  } else if (Period == "FIRST OF SEP") {
    Month = 9
  } else if (Period == "FIRST OF DEC") {
    Month = 12
  } else {
    Month = NULL
  }
  return(Month)
}


# USDA_NASS
chickenInventory <- chickenInventory %>%
  select(Year,Value, Period) %>%
  mutate(Value = Value %>% as.numeric()) %>%
  mutate(Year = Year %>% as.factor()) %>%
  setNames(c("Year", "chickenInventory", "Period")) %>%
  mutate(Month = purrr::map_dbl(Period, Period2Month) %>% as.character()) %>%
  mutate(Year = as.character(Year)) %>%
  mutate(Date = paste(Year, Month, sep = "-")) %>%
```

```
  mutate(Date = lubridate::ymd(Date, truncated = 1L)) %>%
  select(Date, chickenInventory)
hogInventory <- hogInventory %>%
  select(Year,Value, Period) %>%
  mutate(Value = Value %>% as.numeric()) %>%
  mutate(Year = Year %>% as.factor()) %>%
  setNames(c("Year", "hogInventory", "Period")) %>%
  mutate(Month = purrr::map_dbl(Period, Period2Month) %>% as.character()) %>%
  mutate(Year = as.character(Year)) %>%
  mutate(Date = paste(Year, Month, sep = "-")) %>%
  mutate(Date = lubridate::ymd(Date, truncated = 1L)) %>%
  filter(Month == 12) %>%
  select(Date, hogInventory)
cattleInventory <- cattleInventory %>%
  filter(Value != "(D)") %>%
  select(Year,County,Value, `Ag District`, `Ag District Code`) %>%
  mutate(Value = Value %>% as.numeric()) %>%
  mutate(Year = Year %>% as.factor()) %>%
  setNames(c("Year","County","cattleInventory", "AgDistrict", "Code")) %>%
  group_by(Year) %>%
  summarise(cattleInventory = sum(cattleInventory), ncounty= n()) %>%
  filter(ncounty == max(ncounty)) %>% ## remove 1977, because only 26 county reported
  mutate(Year = as.numeric(as.character(Year))) %>%
  mutate(Date = lubridate::ymd(Year, truncated = 2L)) %>%
  select(Date, cattleInventory)
animalInventory <- chickenInventory %>%
  full_join(hogInventory) %>%
  full_join(cattleInventory) %>%
  pivot_longer(-Date, names_to = "animal", values_to = "value") %>%
  mutate(year = lubridate::year(Date)) %>%
  select(-Date) %>%
  group_by(year, animal) %>%
  summarise(value = mean(value, na.rm = TRUE)) %>%
  pivot_wider(names_from= "animal", values_from = "value") %>%
  ungroup %>%
  mutate(state = "iowa") %>%
  select(c("state", "year", "cattleInventory", "hogInventory")) %>%
  mutate(year = lubridate::ymd(year, truncated = 2L)) %>%
  na.omit()
hogInventory <- animalInventory %>% select(state, year, hogInventory)
cattleInventory <- animalInventory %>% select(state, year, cattleInventory)


eggProduction <- eggProduction %>%
  rename(eggProduction = Value) %>%
  select(Year, Period, State, eggProduction) %>%
```

```
  mutate(Period = Period %>% stringr::str_to_title(string = ., locale = "en")) %>%
  mutate(month = Period %>% match(.,month.abb)) %>%
  mutate(day = 1) %>%
  na.omit() %>%
  mutate(period = lubridate::as_date(paste(Year, month, day,sep="-"))) %>%
  mutate(region =  State %>% stringr::str_to_lower(string = ., locale = "en")) %>%
  select(region, period, eggProduction) %>%
  rename(state = region, month = period)

electricGeneration <- electricGeneration %>%
  setNames(c("X1", "year", "month", "state", "X2", "source", "electricGeneration")) %>%
  select(year, month, state, source, electricGeneration) %>%
  filter(state == "IA") %>%
  group_by(year, month, source) %>%
  summarize(electricGeneration = sum(electricGeneration, na.rm = TRUE)) %>%
  mutate(state = "iowa") %>%
  ungroup() %>%
  mutate(month = month %>% lubridate::ymd(truncated = 2)) %>%
  select(state, month, electricGeneration)

electricitySale <- electricitySale %>%
  setNames(c("Month", "electricitySale")) %>%
  separate(col = Month,  c("month", "year")) %>%
  mutate(month = month %>% match(., month.abb) ) %>%
  mutate_at(c("month", "year"), as.numeric) %>%
  mutate(year = ifelse(test = year > 50, yes = year + 1900, no = year + 2000)) %>%
  mutate(month = lubridate::make_date(year, month)) %>%
  mutate(state = "iowa") %>%
  select(state, month, electricitySale)

renewablebiofuel <- renewablebiofuel %>%
  rename(year = Year) %>%
  mutate(state = "iowa") %>%
  mutate(year = lubridate::ymd(year, truncated = 2L)) %>%
  select(state, year, EtOHproduction, biodieselProduction)
EtOHproduction <- renewablebiofuel %>% select(state, year, EtOHproduction)
biodieselProduction <- renewablebiofuel %>% select(state, year, biodieselProduction)

flood <- flood %>%
  select(CZ_NAME_STR, BEGIN_DATE, EVENT_TYPE) %>%
  setNames(c("region", "period", "stormEvent")) %>%
  mutate(region = region %>% stringr::str_replace(" CO.", "")) %>%
  mutate(region = region %>% stringr::str_replace(" \\(ZONE\\)", "")) %>%
  mutate(region = region %>% stringr::str_to_lower(locale = "en")) %>%
  mutate(stormEvent = "flood")
drought <- drought %>%
```

```
  select(CZ_NAME_STR, BEGIN_DATE, EVENT_TYPE) %>%
  setNames(c("region", "period", "stormEvent")) %>%
  mutate(region = region %>% stringr::str_replace(" CO.", "")) %>%
  mutate(region = region %>% stringr::str_replace(" \\(ZONE\\)", "")) %>%
  mutate(region = region %>% stringr::str_to_lower(locale = "en")) %>%
  mutate(stormEvent = "drought")
highWind <- highWind %>%
  select(CZ_NAME_STR, BEGIN_DATE, EVENT_TYPE) %>%
  setNames(c("region", "period", "stormEvent")) %>%
  mutate(region = region %>% stringr::str_replace(" CO.", "")) %>%
  mutate(region = region %>% stringr::str_replace(" \\(ZONE\\)", "")) %>%
  mutate(region = region %>% stringr::str_to_lower(locale = "en")) %>%
  mutate(stormEvent = "highWind")
stormEvent <- flood %>%
  rbind(drought) %>%
  rbind(highWind) %>% # head(1000) %>%
  separate(period, c("month", "day", "year"), "/") %>%
  mutate_at(c("month", "day", "year"), as.numeric) %>%
  mutate(year = ifelse(test = year > 50, yes = year + 1900, no = year + 2000)) %>%
  mutate(period = lubridate::make_date(year, month, day)) %>%
  select(region, period, stormEvent) %>%
  rename(county = region, day = period) %>%
  na.omit() %>%
  distinct() %>%
  mutate(value =1) %>%
  pivot_wider(names_from = stormEvent, values_from = value, values_fill = list(value =
        0)) %>%
  tibble::rowid_to_column() %>%
  mutate(year = lubridate::year(day),
         month = lubridate::month(day)) %>%
  mutate(month = paste(year, month, sep = "-")) %>%
  select(county, month, flood, drought, highWind) %>%
  group_by(county, month) %>%
  summarise_all(sum, na.rm = TRUE) %>%
  ungroup() %>%
  mutate(month = month %>% ymd(truncated = 1L)) %>%
  merge(expand.grid(month = seq(min(.$month), max(.$month), by = "1 month"),
                    county = .$county %>% unique),
        ., by = c("month", "county"),  all.x = TRUE) %>%
  mutate(year = year(month)) %>%
  complete(year, nesting(month,county), fill = list(flood = 0, drought = 0, highWind = 0)) %>%
  select(county, month, flood, drought, highWind)
flood <- stormEvent %>% select(county, month, flood)
drought <- stormEvent %>% select(county, month, drought)
highWind <- stormEvent %>% select(county, month, highWind)
```

```
waterQuality <- waterQuality %>%
  filter(analyte %in% c("Escherichia coli", "Dissolved oxygen (DO)",
                        "Turbidity", "Microcystin")) %>%
  select(county, sampleDate, analyte, result) %>%
  tidyr::separate(col = sampleDate, into = c("month", "day", "year"), convert = TRUE) %>%
  mutate(year = ifelse(test = year > 50, yes = year + 1900, no = year + 2000)) %>%
  mutate(day = paste(year, month, day, sep = "-") %>% lubridate::ymd()) %>%
  select(county, day, analyte, result) %>%
  setNames(c("county", "day", "analyte", "waterQuality")) %>%
  mutate_if(is.character, as.factor) %>%
  group_by(county, day, analyte) %>%
  summarise(waterQuality = waterQuality %>% mean(na.rm = TRUE)) %>%
  ungroup() %>%
  tidyr::pivot_wider(names_from = analyte, values_from = waterQuality,
                     values_fill = list(waterQuality = NA),
                     values_fn = list(waterQuality = mean)) %>%
  setNames(c("county", "day", "Ecoli", "dissolvedOxygen", "microcystin", "turbidity")) %>%
  mutate(county = county %>% stringr::str_to_lower(locale = "en"))
Ecoli <- waterQuality %>% select(county, day, Ecoli)
dissolvedOxygen <- waterQuality %>% select(county, day, dissolvedOxygen)
microcystin <- waterQuality %>% select(county, day, microcystin)
turbidity <- waterQuality %>% select(county, day, turbidity)

fishKill <- fishKill %>%
  select(date, county) %>%
  mutate(fishKill = 1) %>%
  mutate(date = date %>% as.character() %>% stringr::str_sub(end = 10) %>% ymd(truncated =
           1L)) %>%
  rename(day = date) %>%
  select(county, day, fishKill) %>%
  mutate(year = lubridate::year(day),
         month = lubridate::month(day)) %>%
  mutate(month = paste(year, month, sep = "-")) %>%
  select(county, month, fishKill) %>%
  mutate(county = county %>% stringr::str_to_lower(locale = "en")) %>%
  group_by(county, month) %>%
  summarise_all(sum, na.rm = TRUE) %>%
  ungroup() %>%
  mutate(month = month %>% ymd(truncated = 1L)) %>%
  merge(expand.grid(month = seq(min(.$month), max(.$month), by = "1 month"),
                    county = .$county %>% unique),
        ., by = c("month", "county"),  all.x = TRUE) %>%
  mutate(year = year(month)) %>%
  complete(year, nesting(month,county), fill = list(fishKill = 0)) %>%
  select(county, month, fishKill)
```

```
electricityPrice <- electricityPrice %>%
  janitor::row_to_names(row_number = 1) %>%
  mutate_if(is.numeric, signif, digits = 1 ) %>%
  readr::type_convert() %>%
  filter(State == "IA") %>%
  mutate(state = "iowa") %>%
  select(state, Year, Total) %>%
  mutate(Year = Year %>% lubridate::ymd(truncated = 2L)) %>%
  setNames(c("state", "year", "electricityPrice"))

govExpenditure <- govExpenditure %>%
  select(`Budget FY`, `Fiscal Period`, `Amount`) %>%
  setNames(c("year", "month", "govExpenditure")) %>%
  mutate(month = lubridate::ymd(paste(year, month, sep = "-"),  truncated = 1L)) %>%
  group_by(month) %>%
  summarise(govExpenditure = govExpenditure %>% sum(na.rm = TRUE)) %>%
  ungroup() %>%
  mutate(state = "iowa") %>%
  select(state, month, govExpenditure) %>%
  na.omit()
govRevenue <- govRevenue %>%
  select(`Budget FY`, `Fiscal Period`, `Amount`) %>%
  setNames(c("year", "month", "govRevenue")) %>%
  mutate(month = lubridate::ymd(paste(year, month, sep = "-"),  truncated = 1L)) %>%
  group_by(month) %>%
  summarise(govRevenue = govRevenue %>% sum(na.rm = TRUE)) %>%
  ungroup() %>%
  mutate(state = "iowa") %>%
  select(state, month, govRevenue) %>%
  na.omit()

employment <- employment %>%
  select(Year, Quarter, `Area Name`, `Month 1`, `Month 2`, `Month 3`) %>%
  setNames(names(.) %>% stringr::str_remove(pattern = " ")) %>%
  rename(county = AreaName) %>%
  filter(county != "Statewide") %>%
  mutate(county = county %>% stringr::str_to_lower(locale = "en")) %>%
  group_by(Year, Quarter, county) %>%
  summarise(Month1 = sum(Month1, na.rm = TRUE),
          Month2 = sum(Month2, na.rm = TRUE),
          Month3 = sum(Month3, na.rm = TRUE)) %>%
  ungroup() %>%
  pivot_longer(-c(Year, Quarter, county), names_to = "month",  values_to =
          "employment") %>%
  mutate(month = month %>% stringr::str_remove(pattern = "Month") %>% as.integer()) %>%
  mutate(month = month * Quarter) %>%
```

```
  mutate(month = lubridate::ymd(paste(Year, month, sep = "-"), truncated = 1L)) %>%
  select(county, month, employment) %>%
  arrange(month, county)


dieselPrice <- dieselPrice %>%
  janitor::row_to_names(row_number = 1) %>%
  select(Date, starts_with("Midwest")) %>%
  readr::type_convert() %>%
  setNames(c("month", "dieselPrice")) %>%
  na.omit() %>%
  separate(col = month, into = c("month", "year")) %>%
  mutate(month = month %>% match(.,month.abb)) %>%
  mutate(month = paste(year, month, sep = "-") %>% lubridate::ymd(truncated = 1L)) %>%
  mutate(state = "iowa") %>%
  select(state, month, dieselPrice)

# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ polygon_data
        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

period_by <- c("day", "month", "year")
region_by <- c("county_sf", "AgDistrict_sf", "state_sf")

save(eggProduction, # state month # USDA NASS: https://www.nass.usda.gov/
     cornYield, # county  year # USDA NASS: https://www.nass.usda.gov/
     soybeanYield, # county  year # USDA NASS: https://www.nass.usda.gov/
     cornSilage, # AgDistrict year # USDA NASS: https://www.nass.usda.gov/
     cattleInventory, hogInventory, # year state # USDA NASS: https://www.nass.usda.gov/
     cornPrice, soybeanPrice, # state month # ISU Extension and Outreach, Ag Decision Maker
                    https://www.extension.iastate.edu/agdm/crops/html/a2-11.html
     flood, drought, highWind, # county month # NOAA Storm Events Database:
            https://www.ncdc.noaa.gov/stormevents/
     fishKill, # county month # Iowa DNR Fish Kill Database:
            https://programs.iowadnr.gov/fishkill/
     dieselPrice, # state month #EIA Gasoline and Diesel Fuel Update:
            https://www.eia.gov/petroleum/gasdiesel/
     electricGeneration, # state month #  State-level generation and fuel consumption data (EIA-
            923): https://www.eia.gov/electricity/data.php
     electricitySale, # state month # Monthly Form EIA-861M (formerly EIA-826) detailed data:
            https://www.eia.gov/electricity/data.php#sales:
            https://www.eia.gov/electricity/data.php#sales
     electricityPrice, # state year # Annual retail price (EIA-861):
            https://www.eia.gov/electricity/data.php#sales
     EtOHproduction, biodieselProduction, # state year # Iowa Renewable Fuels Association:
            https://iowarfa.org/resource-center/statistics/
```

```
    employment, # county month # Iowa Workforce Development, QCEW:
        https://www.iowaworkforcedevelopment.gov/quarterly-census-employment-and-wages
    govRevenue, # state month $ State of Iowa's data portal: https://data.iowa.gov/State-
        Government-Finance/State-of-Iowa-Revenue/urps-v5ck
    govExpenditure, # state month # State of Iowa's data portal: https://data.iowa.gov/State-
        Government-Finance/State-of-Iowa-Expenditures/mn9y-cwp6
    Ecoli, dissolvedOxygen, microcystin, turbidity, # county day # AQuIA, Iowa DNR Surface
        Water Monitoring data: https://programs.iowadnr.gov/aquia/
    file = "data/polygon_data.RData")

load("data/polygon_data.RData", verbose = TRUE)
```

APPENDIX C: CODE FOR WATER DATA QUERYING

```
setwd(dirname(rstudioapi::getSourceEditorContext()$path))
rm(list=ls())
dev.off()

library(dataRetrieval)
library(lubridate)
library(dplyr)
library(purrr)
library(furrr)
library(tidyr)
library(sf)
source("ud_function.r")

# query site information for state around IA
states <- c("NE", "IA", "KS", "MO", "IL", "SD", "MN", "WI")
siteMidWest_info <- read_siteMidWest_info(states = states, update =FALSE)

# rbind site information only for select parm_cd
siteMidWest_parm <- rbind(siteMidWest_info %>%
                          filter(end_date > "2019-10-31") %>%
                          filter(data_type_cd == "dv") %>%
                          filter(parm_cd %in% c("00060", "00010", "99133", "00065")) %>%
                          filter(stat_cd == "00003"),
                         siteMidWest_info %>%
                          filter(end_date > "2019-12-31") %>%
                          filter(data_type_cd == "qw") %>%
                          filter(parm_cd %in% "80154"))

# convert site dataframe to sf
siteMidWest_sf <- site2sf(df=siteMidWest_parm,
                         id_cn="site_no",
                         Lon_cn="dec_long_va",
                         Lat_cn="dec_lat_va",
                         crs= 4326)

buffer_dist <- 10000

# import HUC 07 10, buffering
HU2_bf <- st_union(sf::st_read("data/HUC_IA/WBD_07_HU2_Shape/WBDHU2.shp", quiet =
           F),
              sf::st_read("data/HUC_IA/WBD_10_HU2_Shape/WBDHU2.shp", quiet = F)) %>%
  thinshp() %>%
  st_transform(crs = 2163) %>%
```

```
  st_buffer(dist = buffer_dist) %>%
  st_transform(crs = 4326)

# import boundary, convert sf, buffering
IAMO_sf_bf <- sf::st_as_sf(maps::map("state",
                                     plot = FALSE,
                                     fill = TRUE)) %>%
  # filter(ID %in% c("iowa","missouri")) %>%
  filter(ID %in% states) %>%
  st_transform(crs = 2163) %>%
  st_buffer(dist = buffer_dist) %>%
  st_transform(crs = 4326)

# intersection between site and boundary with buffering
siteMidWest_select_sf <- siteMidWest_sf %>%
  #st_intersection(IAMO_sf_bf) %>%
  st_intersection(HU2_bf)

ggplot() +
  #geom_sf(data = IAMO_sf_bf) +
  geom_sf(data = HU2_bf) +
  geom_sf(data = siteMidWest_select_sf)

siteMidWest_select <- siteMidWest_parm %>%
  filter(site_no %in% siteMidWest_select_sf$id); siteMidWest_select

# remove unnessary datastream
siteMidWest_select <- siteMidWest_select %>%
  filter(site_no !="05420500" | parm_cd != "00010" | ts_id !=42791) %>%
  filter(site_no !="05420500" | parm_cd != "99133" | ts_id !=230012) %>%
  filter(site_no !="05544385" | parm_cd != "00065" | ts_id !=155322) %>%
  filter(site_no !="06485950" | parm_cd != "00065" | ts_id !=247531) %>%
  filter(site_no !="06903900" | parm_cd != "00060" | ts_id !=43342) %>%
  filter(site_no !="424848088083100" | parm_cd != "00065" | ts_id !=155613)

siteMidWest_select <- siteMidWest_select %>%
  filter(site_no !="411219096010601") %>%
  filter(site_no !="05536995") %>%
  filter(site_no !="05536137") %>%
  filter(site_no !="05340500") %>%
  filter(site_no !="05357206") %>%
  filter(site_no !="05398000") %>%
  filter(site_no !="473423095053301") %>%
  filter(site_no !="054279465")

siteMidWest_select <- siteMidWest_select %>% filter(!is.na(stat_cd))
```

```
# check for parm_cd
siteMidWest_select %>% group_by(stat_cd) %>% summarise(n=n()) %>% ungroup() %>%
        arrange((desc(n)))
# check for parm_cd
siteMidWest_select %>% group_by(parm_cd) %>% summarise(n=n())  %>% ungroup() %>%
        arrange((desc(n)))
# check duplicate sit_no in data stream
siteMidWest_select %>% group_by(site_no) %>% summarise(n=n()) %>% ungroup() %>%
        arrange((desc(n)))
# check for each lake each parm_cd
siteMidWest_select %>% group_by(site_no, parm_cd) %>% summarise(n=n()) %>%
        ungroup() %>% arrange((desc(n)))
# check a specific site
# a <- siteMidWest_select %>% filter(site_no == "424848088083100");View(a)
# a <- siteMidWest_select %>% filter(parm_cd == "00010") ;View(a)

# save selected site info
save(siteMidWest_select,
    file = "data/siteMidWest_select.Rdata")
load("data/siteMidWest_select.Rdata", verbose = TRUE)

# pulling water quality data from online:
library(tictoc)
tic()
pCodes <- siteMidWest_select$parm_cd %>% unique(); pCodes
water_data <- plyr::llply(pCodes, function(pCodes) {
  querydatas(pCode= pCodes,
          site = siteMidWest_select,
          startDate = "1980-01-01",
          endDate = "2019-12-31") %>%
    left_join(siteMidWest_select %>% # add long lat
            select(site_no, dec_long_va, dec_lat_va) %>%
            distinct(),
           by = "site_no") %>%
    select(site_no, dec_long_va, dec_lat_va, parm_cd, data_type_cd, stat_cd, data) %>%
    rename(longitude = dec_long_va, latitude = dec_lat_va) %>%
    mutate(isnull = map_dbl(data, is_null)) %>%  # remove null stie
    filter(isnull==0) %>%
    select(-isnull) %>%
    mutate(site_no = paste0("USGS", site_no))
}); water_data
toc()


# parameterCdFile %>% filter(parameter_cd== "80154")
```

```
# 00010: Temperature, water, degrees Celsius
# 00060: Stream flow (Discharge), mean. daily, cubic feet per second
# 99133: Inorganic nitrogen (nitrate and nitrite) in situ, milligrams per liter as nitrogen
# 80154: Suspended sediment concentration, milligrams per liter
# 00065: Gage height, feet

# 00060: charge, cubic feet per second, Stream flow, mean. daily
wflow <- water_data[[1]] %>%
  unnest(cols = c(data)) %>%
  select(site_no, longitude, latitude, sample_dt, result) %>%
  setNames(c("id", "longitude", "latitude", "Date", "wflow")) %>%
  mutate(wflow = purrr::map_dbl(wflow, function(x) if_else(x < 0, 0, x)))

# 00065: Gage height, feet
gageHeight <- water_data[[2]] %>%
  unnest(cols = c(data)) %>%
  select(site_no, longitude, latitude, sample_dt, result) %>%
  setNames(c("id", "longitude", "latitude", "Date", "gageHeight"))

# 00010: Temperature, water, degrees Celsius, Temperature, water
twater<- water_data[[3]] %>%
  unnest(cols = c(data)) %>%
  select(site_no, longitude, latitude, sample_dt, result) %>%
  setNames(c("id", "longitude", "latitude", "Date", "twater")) %>%
  mutate(twater = twater + 273.15) %>%
  filter(twater > 0)

# 99133: Nitrate plus nitrite, water, in situ, milligrams per liter as nitrogen
nitrateNitrite <- water_data[[4]] %>%
  unnest(cols = c(data)) %>%
  select(site_no, longitude, latitude, sample_dt, result) %>%
  setNames(c("id", "longitude", "latitude", "Date", "nitrateNitrite"))

# 80154: Suspended sediment concentration (SSC), milligrams per liter
SSC <- water_data[[5]] %>%
  unnest(cols = c(data)) %>%
  select(site_no, longitude, latitude, sample_dt, result) %>%
  setNames(c("id", "longitude", "latitude", "Date", "SSC"))


save(water_data,
     wflow, gageHeight, twater, nitrateNitrite,
     file = "data/water_data.Rdata")
load("data/water_data.Rdata", verbose = TRUE)
```

APPENDIX D: CODE FOR WEATHER DATA QUERYING

```
setwd(dirname(rstudioapi::getSourceEditorContext()$path))
rm(list=ls())
dev.off()

library(tidyr)
library(rnoaa)
library(dplyr)
library(lubridate)
library(sf)
# stationUSA_info <- ghcnd_stations()
# save(stationUSA_info, file = "data/stationUSA_info.Rdata")
load("data/stationUSA_info.Rdata")
source("ud_function.R")

# subset the stationUSA_info by state, first_year, and last_year
station_info <- stationUSA_info %>%
  filter(element %in% c("PRCP ", "TMAX", "TMIN")) %>%
  filter(state %in% c("IA", "MO")) %>%
  filter(first_year < "1980") %>%
  filter(last_year >= "2019");  # head(station_info)

# pulling data from online:
# list of avaliable variable from query:
        https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/readme.txt
tictoc::tic()
monitors <- station_info$id
weather_data <- meteo_pull_monitors(monitors, var = c("prcp", "tmax", "tmin"),
                                    date_min = "1980-01-01",
                                    date_max = "2019-12-31") %>%
  rename(Date = date, NOAA_ID = id)
tictoc::toc()

# remove the all() NA raw
rmRow <- weather_data %>%
  select(prcp, tmax, tmin) %>%
  is.na() %>%
  apply(MARGIN = 1, FUN = all)
weather_data <- weather_data[!rmRow,]


weather_data<- weather_data %>% left_join(station_info %>%
                                  select(id, latitude,longitude) %>%
                                  distinct,
```

```
                              by = c("NOAA_ID" = "id")) %>%
  select(NOAA_ID, longitude, latitude, Date, prcp, tmax, tmin); weather_data


prcp <- weather_data %>%
  select(NOAA_ID, longitude, latitude, Date, prcp) %>%
  setNames(c("id", "longitude", "latitude", "Date", "prcp")) %>%
  mutate_at("prcp", function(x) x/10)

tmax <- weather_data %>%
  select(NOAA_ID, longitude, latitude, Date, tmax) %>%
  setNames(c("id", "longitude", "latitude", "Date", "tmax")) %>%
  mutate_at("tmax", function(x) x/10) %>%
  mutate(tmax = tmax + 273.15) %>%
  filter(tmax > 0)

tmin <- weather_data %>%
  select(NOAA_ID, longitude, latitude, Date, tmin) %>%
  setNames(c("id", "longitude", "latitude", "Date", "tmin")) %>%
  mutate_at("tmin", function(x) x/10) %>%
  mutate(tmin = tmin + 273.15) %>%
  filter(tmin > 0)

# save weather data
save(weather_data,
     prcp, tmax, tmin,
     file = "data/weather_data.Rdata")
load("data/weather_data.Rdata", verbose = TRUE)
```

APPENDIX E: CODE FOR DATA INTEGRATION AND MODELING

```
setwd(dirname(rstudioapi::getSourceEditorContext()$path))
rm(list=ls())
dev.off()
options(future.globals.maxSize = 600*1024^2) # 600 Mb

# load pkg
## data modification
library(tibble)
library(dplyr)
library(tidyr)
library(data.table)
library(lubridate)
library(sf)
library(stringr)
library(forcats)
## utility
library(purrr)
library(furrr)
library(tictoc)
## visuliation
library(ggplot2)
library(gRain)
library(Rgraphviz)
library(igraph)
# modeling
library(bnlearn)
library(XMRF)

# load function
source("ud_function.R")

# load data
load("data/region_sf.RData", verbose = TRUE)
load("data/water_data.Rdata", verbose = TRUE)
load("data/weather_data.Rdata", verbose = TRUE)
load("data/polygon_data.RData", verbose = TRUE)
rm(weather_data, water_data)
dataSummary <- readr::read_csv("dataSummary.csv") %>%
  select("Variable", "Space Scale", "Time Scale", "Spatial downscaling", "Temporal
        downscaling") %>%
  setNames(c("var", "orig_region", "orig_period", "Spatialdownscaling",
        "Temporaldownscaling"))
```

```
# set control boundary
boundary <- "ia_sf"

# crop point data by the base boundary
point_data <- c("prcp", "tmax", "tmin", "twater", "wflow", "gageHeight")
for (i in 1:length(point_data)) assign(point_data[i], point_data[i] %>%
                                  get() %>%
                                  crop_data_by_boundary(boundary = boundary))
# rescale
for (i in 1:length(point_data)) assign(point_data[i], point_data[i] %>%
                                  get() %>%
                                  rescale_point_data(x= ., to = c(1, 5)))

# crop the all geo-boundary by the base boundary
regions <- c("county_sf", "AgDistrict_sf")
for (i in 1:length(regions)) assign(regions[i], crop_region_by_boundary(region_sf = regions[i],
                                                     boundary = boundary))
periods <- c("day", "month", "year")
regions <- c("county_sf", "AgDistrict_sf", "state_sf")
# Visialize regional division
ggplot() +
  geom_sf(data = county_sf, aes(fill = region), alpha = 0, size =0.5, color = "black") +
  geom_sf(data = AgDistrict_sf, aes(fill = region), alpha = 0, size = 1, color = "black") +
  geom_sf(data = ia_sf, aes(fill = region), alpha = 0, size = 1, color = "black") +
  theme(legend.position = "none") +
  ggtitle("Regional division of Iowa by Agriculture Distrist and county")

# downscale spatiotemporal data by variables
## point_data
initialDataSummary <- dataSummary %>%
  filter(orig_region == "point") %>%
  filter(var != "twater")
### initial joined dataset
initial_pointData_downscaling(point_datas = initialDataSummary$var,
                              periods = periods,
                              regions = regions,
                              Spatial_downscalings = initialDataSummary$Spatialdownscaling,
                              Temporal_downscalings = initialDataSummary$Temporaldownscaling)
### other point data
pointdataSummary <- dataSummary %>%
  filter(var == "twater")
# rm(list = ls(pattern = "pointdata_"))
point_data_downscaling(point_datas = pointdataSummary$var,
                       periods = periods,
                       regions = regions,
                       Spatial_downscalings = pointdataSummary$Spatialdownscaling,
```

```
                    Temporal_downscalings = pointdataSummary$Temporaldownscaling)

## polygon_data
### construcutre original scale and find the avaiable downscale, save as a dict
period_division <- tibble() %>%
  rbind(tibble(from = "year", to = "month")) %>%
  rbind(tibble(from = "year", to = "day")) %>%
  rbind(tibble(from = "month", to = "day"))
region_division <- tibble() %>%
  rbind(tibble(from = "AgDistrict", to = "county")) %>%
  rbind(tibble(from = "state", to = "county")) %>%
  rbind(tibble(from = "state", to = "AgDistrict"))
polygon_data_dict <- dataSummary %>%
  filter(orig_region != "point") %>%
  mutate(dest_region = orig_region %>% purrr::map(function(division_by){
    region_division %>%
      filter(to == division_by) %>%
      select(from) %>%
      rbind(division_by) %>%
      unlist %>% paste()
  })) %>%
  unnest(dest_region) %>%
  mutate(dest_period = orig_period %>% purrr::map(function(division_by){
    period_division %>%
      filter(to == division_by) %>%
      select(from) %>%
      rbind(division_by) %>%
      unlist %>% paste()
  })) %>%
  unnest(dest_period) %>%
  mutate_all(as.character) %>%
  mutate(identical_scale = ifelse(orig_period == dest_period & orig_region == dest_region,
          TRUE, FALSE))
polygon_data_downscaling(polygon_data_dict = polygon_data_dict,
                         regional_division_df = regional_division_countyAgDistrictstate)


## modeling at multiscale
### create joined data and model
# test: period <- periods[2]; region <- regions[3]
rm(list = ls(pattern = "joined|model"))
for (period in periods){
  for (region in regions){
    set.seed((10))
    scale_label <- paste0(stringr::str_sub(region, end = -4L), "X", period)
```

```
    # create joined area and point data
    cat("\n\n"); cat(paste0("joined_", scale_label))
    tmp_joined <- forwardSelection(scale_label, rrp = 0.8) %>%
      mutate(period = as.factor(period)) %>%
      mutate(region = as.factor(region)) %>%
      select(region, period, everything()) %>%
      arrange(region, period) %>%
      # mutate_if(is.numeric, function(x) log2(x+1)) %>% # log transform
      # mutate_if(is.numeric, scales::rescale, to = c(1, 5)) # %>% # rescaling
      mutate_at(c("region", "period"), as.character) %>%
      mutate_if(is.numeric, arules::discretize, method = "interval", breaks = 5) %>%  # discretize
      mutate_if(is.factor, as.numeric) # discretize # %>% add_time_space_varible(scale_label =
          scale_label)

    # tmp_joined %>% summary() # check

    # save joined
    tmp_joined %>%
      assign(value = .,
             x =  paste0("joined", "_", scale_label),
             envir = .GlobalEnv)

    # rm na
    tmp_joined <- tmp_joined %>%
      arrange(region, period) %>%
      na.omit()
    cat(paste0(" (Dim: ", nrow(tmp_joined), " ", ncol(tmp_joined), ")"))

    # create model and strength
    if (nrow(tmp_joined)!=0){
      cat("\n"); cat("model ")
      tmp_model <- tmp_joined %>%
        xmrfModeling(.)
      tmp_model %>%
        assign(value = ., x = paste0("model", "_", scale_label), envir = .GlobalEnv)
    }
  }
}
ls(pattern = "pointdata_|joined_|model_|strength_")
save(list = ls(pattern = "pointdata_|joined_|model_|strength_"),
     file = "data/join_model_strength.Rdata")
# load("data/join_model_strength.Rdata", verbose = TRUE)

# visualization
dev.off()
# test: period <- periods[2]; region <- regions[3]
```

```
par(mfrow=c(3,3), mar=rep(1,4)) ## plot all 9 submodels
for (period in periods){
  for (region in regions){
    set.seed((15))
    scale_label <- paste0(stringr::str_sub(region, end = -4L), "X", period)
    plotMatrix_xmrf(scale_label)
  }
}

# Summarise Networks
networkSummary <- ls(pattern = "model_", envir = .GlobalEnv) %>%
  stringr::str_sub(start=7) %>%
  tibble(Scale = .) %>%
  mutate(networkSummary = Scale %>% map(SummariseNetwork)) %>%
  unnest(networkSummary) %>%
  separate(col = Scale, into = c("Space scale", "Time scale"), sep = "X" ) %>%
  mutate_if(is.character, as.factor) %>%
  mutate(`Space scale` = fct_relevel(`Space scale`, "county", "AgDistrict", "state")) %>%
  mutate(`Time scale` = fct_relevel(`Time scale`, "day", "month", "year")) ; networkSummary
write.csv(networkSummary %>% mutate_if(is.numeric, signif, digits = 3), file =
            "networkSummary.csv")
networkSummary %>%
  psych::pairs.panels(., stars=TRUE, density = FALSE, ellipses=FALSE)


ggpubr::ggarrange(
  ggpubr::ggarrange(combineEdge(by_scale = "spaceScale") + xlab(NULL) + ylab(NULL),
                    combineEdge(by_scale = "timeScale") + xlab(NULL) + ylab(NULL) ,
                    ncol = 2, nrow = 1),
  combineEdge(by_scale = "none") + xlab(NULL) + ylab(NULL),
  ncol = 1, nrow = 2)


ggpubr::ggarrange(ggpubr::ggarrange(networkSummary %>%
                                      ggplot(aes(x= `Time scale`, y = `# of Variable`)) +
                                      geom_point(aes(shape = `Space scale`)) +
                                      geom_line(aes(group = `Space scale`)) +
                                      ylab("# of selected variables") +
                                      xlab(NULL) +
                                      theme_bw(),
                                    networkSummary %>%
                                      ggplot(aes(x= `Time scale`, y = `# of Data point`)) +
                                      geom_point(aes(shape = `Space scale`)) +
                                      geom_line(aes(group = `Space scale`)) +
                                      ylab("# of datapoint") +
                                      xlab("Time Scale") +
```

```
                          theme_bw() +
                          theme(legend.position = c(.91, 0.67)),
                        labels = NA, common.legend = TRUE, legend = "top",
                        ncol = 1, nrow = 2),
              ggpubr::ggarrange(networkSummary %>%
                          ggplot(aes(x= `Space scale`, y = `# of Variable`)) +
                          geom_point(aes(shape = `Time scale`)) +
                          geom_line(aes(group = `Time scale`)) +
                          ylab(NULL) +
                          xlab(NULL) +
                          theme_bw(),
                        networkSummary %>%
                          ggplot(aes(x= `Space scale`, y = `# of Data point`)) +
                          geom_point(aes(shape = `Time scale`)) +
                          geom_line(aes(group = `Time scale`)) +
                          ylab(NULL) +
                          xlab("Space Scale") +
                          theme_bw() +
                          theme(legend.position = c(.91, 0.67)),
                        labels = NA, common.legend = TRUE, legend = "top",
                        ncol = 1, nrow = 2),
              labels = "AUTO", ncol = 2, nrow = 1)

# degree Summary by group
ls(pattern = "model_", envir = .GlobalEnv) %>%
  stringr::str_sub(start=7) %>%
  map(degreeNetork) %>%
  reduce(rbind) %>%
  separate(col = Scale, into = c("Space scale", "Time scale"), sep = "X" ) %>%
  mutate(`Space scale` = fct_relevel(`Space scale`, "county", "AgDistrict", "state")) %>%
  mutate(`Time scale` = fct_relevel(`Time scale`, "day", "month", "year")) %>%
  ggplot(aes(x = Degree)) +
  geom_bar() +
  facet_grid(`Time scale` ~ `Space scale`)

# betweenness Summary by group
ls(pattern = "model_", envir = .GlobalEnv) %>%
  stringr::str_sub(start=7) %>%
  map(betweennessNetork) %>%
  reduce(rbind) %>%
  separate(col = Scale, into = c("Space scale", "Time scale"), sep = "X" ) %>%
  mutate(`Space scale` = fct_relevel(`Space scale`, "county", "AgDistrict", "state")) %>%
  mutate(`Time scale` = fct_relevel(`Time scale`, "day", "month", "year")) %>%
  mutate(`Group` = fct_relevel(`Group`, "Food", "Water", "Weather", "Energy",
        "Economic","Event")) %>%
  ggplot() +
```

```
  geom_boxplot(aes(x= Group,
                y = `Betweenness`)) +
  facet_grid(`Time scale` ~ `Space scale`) +
  theme_bw() +
  xlab(NULL)+
  theme(axis.text.x = element_text(angle = -30))


ggpubr::ggarrange(networkSummary %>%
                    ggplot(aes(x= `Time scale`, y = `Assortativity Coefficient`)) +
                    geom_point(aes(shape = `Space scale`)) +
                    geom_line(aes(group = `Space scale`)) +
                    ylab(NULL) +
                    xlab(NULL) +
                    theme_bw(),
                  networkSummary %>%
                    ggplot(aes(x= `Space scale`, y = `Assortativity Coefficient`)) +
                    geom_point(aes(shape = `Time scale`)) +
                    geom_line(aes(group = `Time scale`)) +
                    ylab(NULL) +
                    xlab(NULL) +
                    theme_bw(),
                  labels = NA, common.legend = FALSE, legend = "top",
                  ncol = 2, nrow = 1)

### normalizty test for
apply(joined_countyXdate %>% select_if(is.numeric), 2, shapiro.test)
apply(joined_countyXdate %>% select_if(is.numeric), 2, nortest::ad.test)

### correlation, density at each scale
#  %>% mutate_if(is.numeric, log2)
psych::pairs.panels(joined_countyXday, stars=TRUE)
psych::pairs.panels(joined_AgDistrictXday, stars=TRUE)
psych::pairs.panels(joined_stateXday, stars=TRUE)
psych::pairs.panels(joined_countyXmonth, stars=TRUE)
psych::pairs.panels(joined_AgDistrictXmonth, stars=TRUE)
psych::pairs.panels(joined_stateXmonth, stars=TRUE)
psych::pairs.panels(joined_countyXyear, stars=TRUE)
psych::pairs.panels(joined_AgDistrictXyear, stars=TRUE)
psych::pairs.panels(joined_stateXyear, stars=TRUE)

joined_countyXday %>% .[,-c(1,2)] %>% as.matrix() %>% psych::pairs.panels(stars=TRUE)
```

```
# Regional division of Iowa by Agriculture Distrist and county
var_sf <- point_data %>%
  purrr::map(function(x){
    get(x =x, envir = .GlobalEnv) %>%
      site2sf() %>%
      select(geometry) %>%
      unique() %>%
      mutate(variable = x)}
  ) %>%
  purrr::reduce(rbind) %>%
  select(variable, geometry)
nvar_sf <- var_sf %>%
  sf::st_intersection(county_sf) %>%
  sf::st_drop_geometry() %>%
  group_by(region) %>%
  distinct() %>%
  summarise(nvar = n()) %>%
  right_join(county_sf, by = "region") %>%
  mutate(nvar = as.factor(nvar)) %>%
  st_as_sf()
nvar_sf %>%
  ggplot() +
  geom_sf(aes(fill = nvar), size =0.5, color = "black") +
  scale_fill_brewer(palette = "Blues") +
  geom_sf(data = st_union(AgDistrict_sf, by_feature = TRUE), alpha = 0, size = 1, color =
          "black") +
  geom_sf(data = var_sf, alpha = 1/6) +
  theme(panel.background = element_blank())
nvar_sf %>% st_drop_geometry() %>% filter(nvar == 6)
```