Project Report

Grant Parajuli, gap6

# Executive Summary

In this project, I focused on downscaling coarse temperature data over Texas from the ERA-5 Reanalysis to finer precipitation data over Texas from NOAA’s CPC Gauge-Based analysis data. I used 2 methods: PCA with KNN, and PCA with linear regression. Notably, these two methods accomplished slightly different aims, as PCA with KNN could be used for the Texas-scale reconstructions at any timesteps within the dataset. The linear regression could instead be used for current or future predictions, but could only predict an individual gridcell value of precipitation.

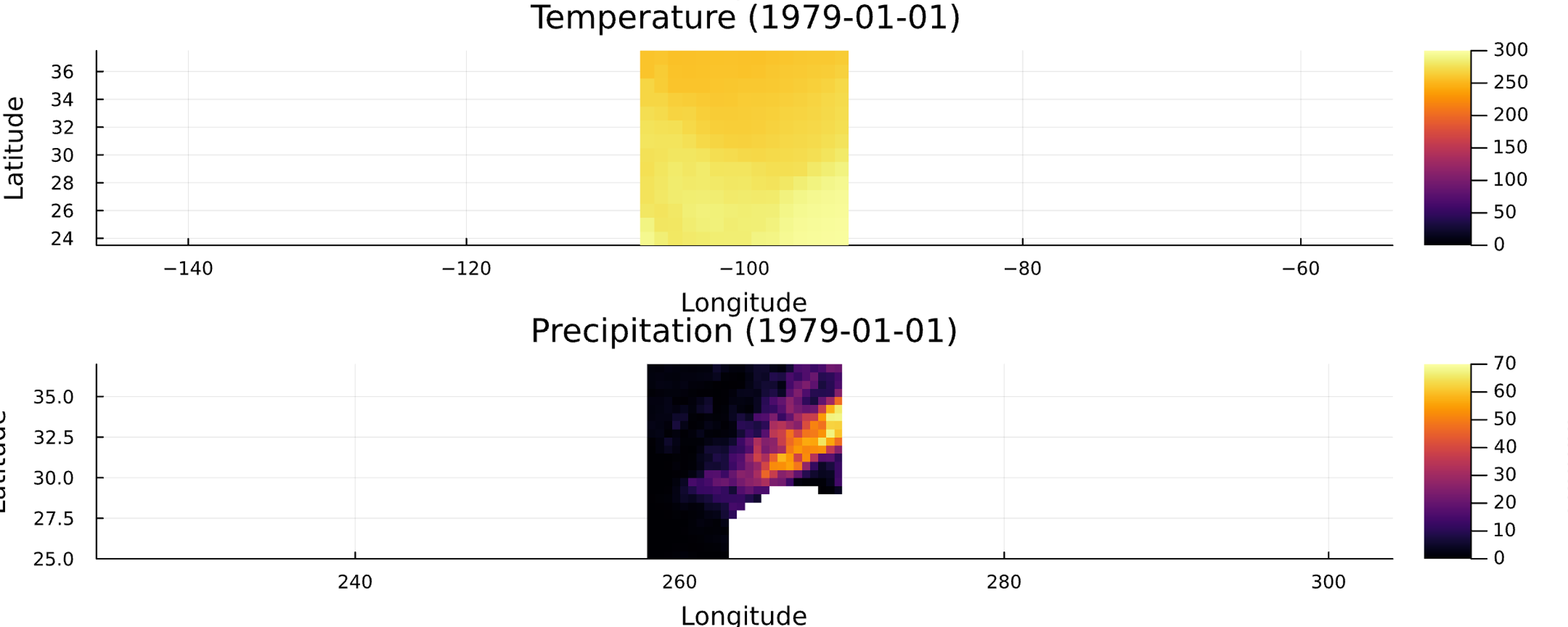
Broadly, the downscaling I’m doing mostly comes from using coarse temperature data to predict finer precipitation data. I mostly used the entire Texas region for this analysis, except for the linear regression that required only a single gridcell value. I used the years 1979-2022 to make sure I had a good sample for both the training and testing data.

The results of applying these methods were mostly inconclusive, as both models were pretty bad for actually predicting precipitation based only on temperature. Multiple improvements could be made for these analyses, including a more statiscally based linear model with a better fit to the data, or adding more variables to condition on to both analyses such as pressure levels/wind speeds. This report would also benefit from a more robust analysis of MSE between the two datasets, but this would require some major code refactoring to be able to run both methods for a large amount of samples and examine the distribution of results.

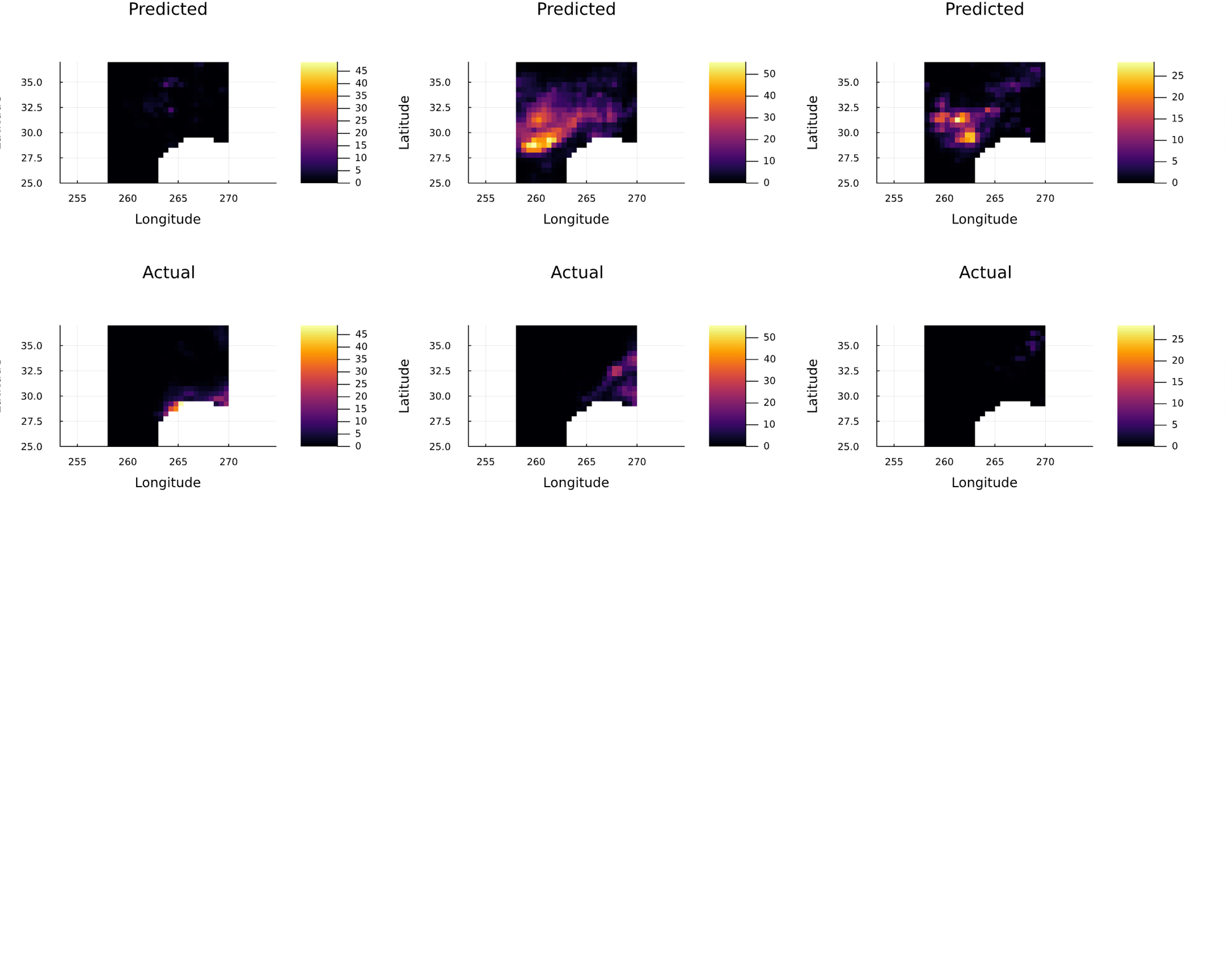
In conclusion, this is a difficult problem for which a very sophisticated model should be considered for a very specific problem. Here, I sought to showcase two different methods of downscaling that may have somewhat different applications. However, if I had a more applied focus, I would pick only one of the models and seek to make it as good as possible for that given application.

# Exploratory Data Analysis

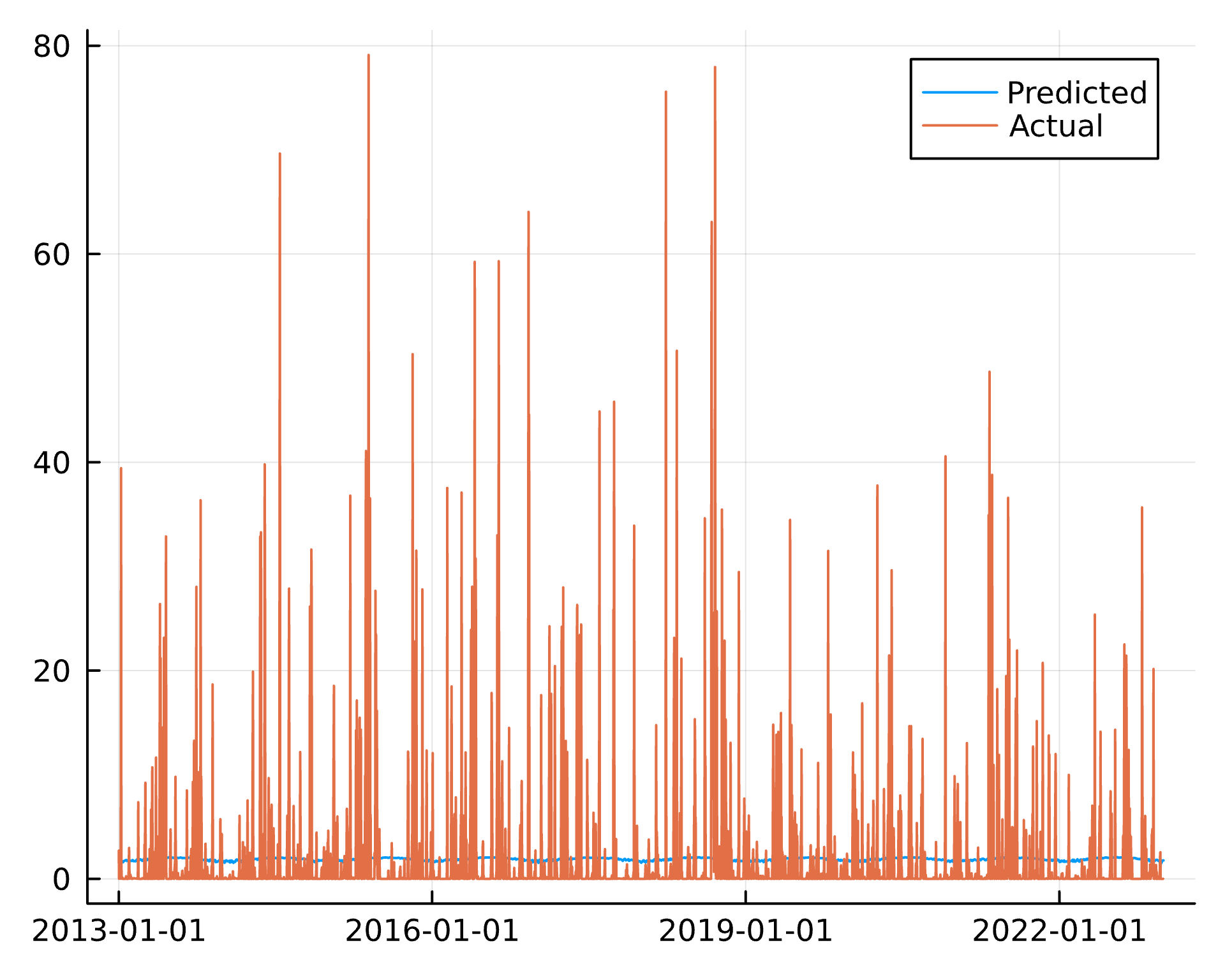
First, here’s a look of a heatmap of both my datasets at a given timestep. One can see that the temperature data is at a coarser resolution than the precipitation data from these plots.



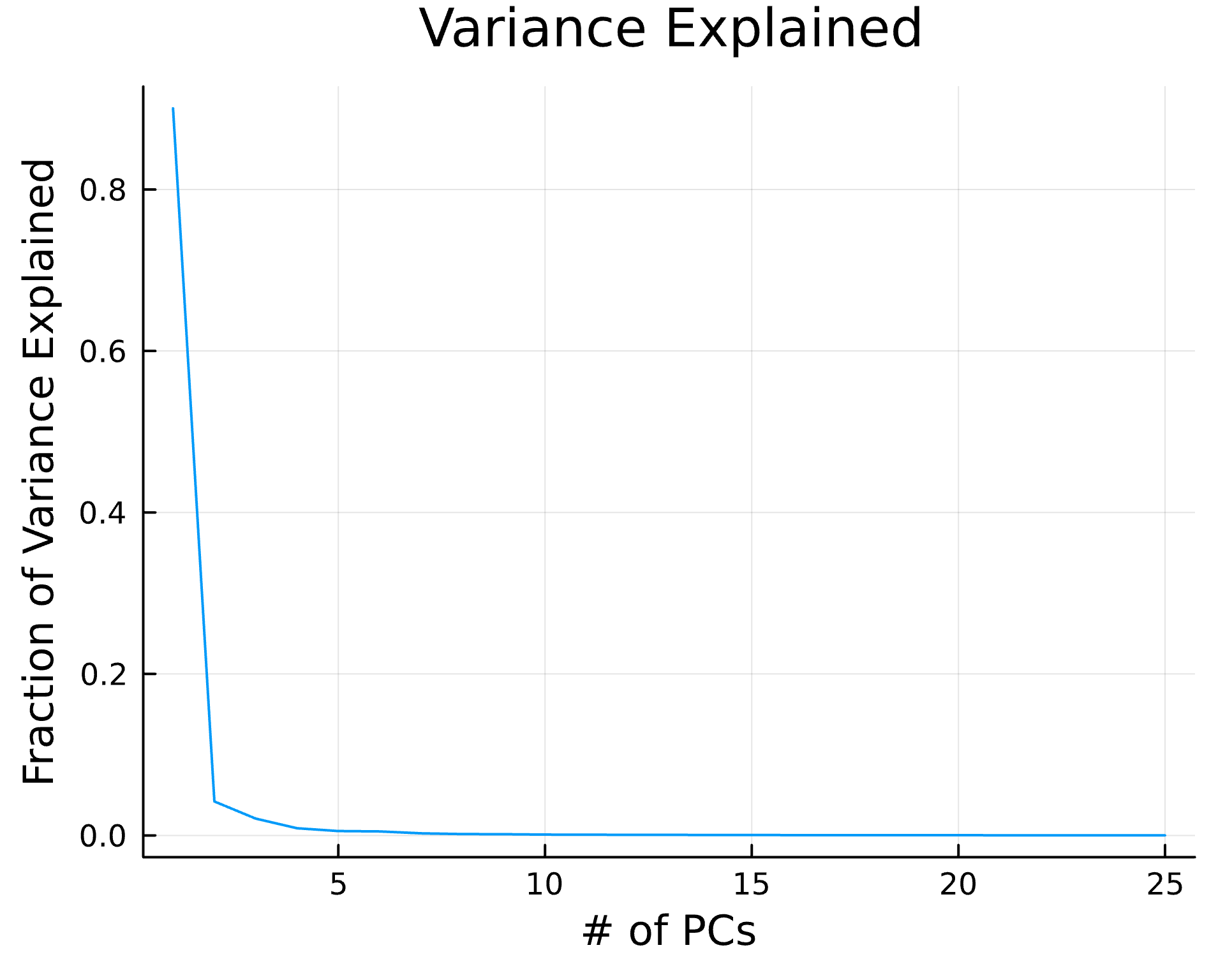
Now, I’ll do a PCA-KNN analysis like we did in Lab 6. I’ll detail the methods more in the relevant section. Below is the results from 3 randomly selected days of this PCA-KNN analysis.



Here is a plot for linear regression. I was able to implement an approach that did linear regression on the principal components to a randomly selected gridcell in the precipitation data.

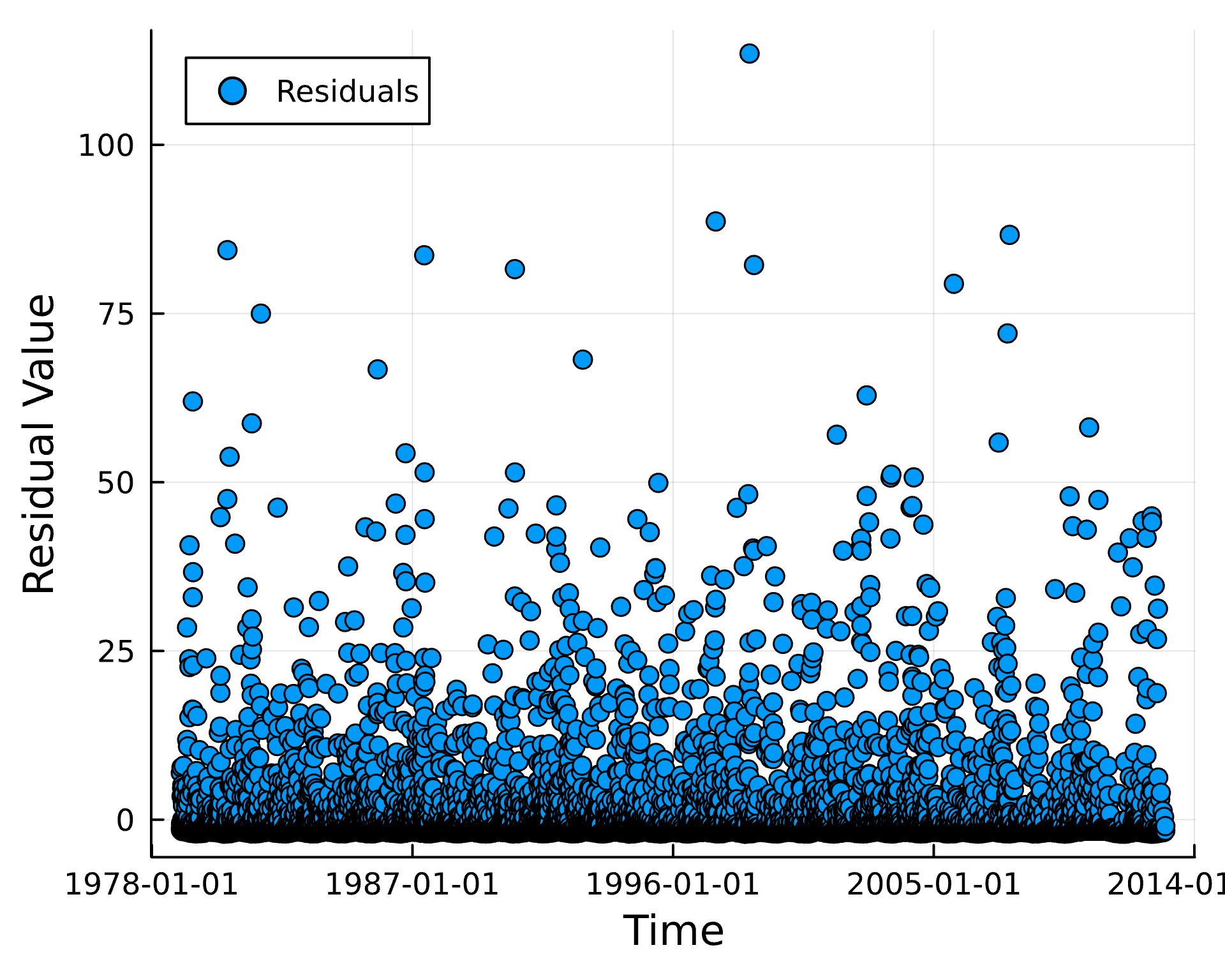


Here are some metrics that see how good my fit was for these examples, and also justify the number of PCA components.



Notice the elbow of the scree plot is pretty much at PC=2. The first principle component explains over 90% of the variance of the data, so using only 2 PCs for KNN is a pretty valid choice as is using only 1 PC for the regression for ease of coding.

Here are the residuals for my linear fit. I’ll comment on these in the model comparison section, but it’s pretty apparent that they’re correlated with my x values.



Next I’ll calculate the difference in MSE between the KNN method and the linear model (postive means that the KNN model has a higher MSE).

50.62003420367596

I calculated MSE for PCA-KNN by averaging it for each of the 3 days shown. Sometimes, the MSE is higher for the PCA-KNN approach, or the MSE might be higher for the linear regression approach. This varies based on the gridcell in precipitation that I’m running the linear model on or the time slice of precipitation I’m running PCA-KNN on. It’s hard to quantify the true MSE of each of these unless I did comprehensive sampling of the data at each gridcell or timestep. This could be a source of improvement on this report, but I struggled mightily to implement the simple cases herein and doing a more comprehensive quantification of MSE would be beyond my current scope.

# Methods

In both methods, I used PCA on the temperature data to reduce the dimensions down from space and time just to principle component space. This was the overarching theme between both KNN and linear regression, as I found this was the most powerful way to do “downscaling” by removing the dimensions of the coarse temperature data, thus allowing me to conduct some kind of modeling with the finer precipitation data. For PCA, I used a train/test split similar to the last lab, where the last 10 years were my testing data. I used data from 1979-2022, as I wanted to have at least 30 years of training data. Temperature data was resampled from hourly data to daily data to match the daily precipitation data.

## PCA-KNN

This method used a non-parametric approach with several steps. In this case, I used KNN in the principal components space to choose the most important time steps from the temperature data. I then used that to select the corresponding time steps in the precipitation data, thus mapping coarse temperature data to finer precipitation data. This process was done for 3 random days to choose as the X\_i in KNN. As evidenced by the output graphs, the actual precipitation values were often very different from the others. In terms of hyperparameter choices, I made a few decisions. The scree plot showed a significant elbow in the first couple principal components, so I deemed it reasonable to just use 3. I used a modest K-value of 3 to decrease the model bias, but this also caused high variance between runs on different days.

## Linear Regression

This method used the Julia package, GLM.jl, to create a generalized linear model between the temperature data principal components and a specific gridcell of the precipitation data. The temperature data principal components allow for the difference in dimensions between the datasets to be irrelevant; however, the major limitation of only examining one gridcell limits how useful this method is. Additionally, this linear model does not use the Bayesian approaches that we learned in class based on log likelihood or a priori estimates, and as far as I can tell, also has little to do with the distributions of our data. I attempted to implement a more statistics-based approach, but I couldn’t solve the multitude of errors it brought, so I was forced to choose a simpler linear model. Given precipitation’s extreme non-linearity and also its tendencies to have values of 0, this generally had a very high MSE/obviously did not fit the data well at all.

# Model Comparison

As we can see from the MSE changing wildly for each model, it’s hard to say which model is better. I’d instead consider the use cases for each model. The KNN approach is exceptionally useful for analyzing the difference at a given time on a regional scale between reconstructed precipitation and the actual precipitation measurements. Given that we’re only conditioning on temperature, which is only a piece of the story of precipitation, I’d expect it not to reconstruct it very well. However, this could be useful to constrain at least some effect of temperature on regional precipitation, even when we have only a coarse regional temperature data product.

The use case for linear regression is also something I see as a big advantage. Precipitation is controlled by many factors, and if more are included as predictors, it’s possible that the regression could become more robust. Being able to predict precipitation trend at a more local location is a valuable tool for those who might live there or a company interested in a specific location. It would be interesting to analyze gauge data within the smaller gridcells and see how it compares to both the observed gridcell value and the predicted gridcell value. A major limitation, however, is the inability to properly simulate extreme values. A better linear regression such as a polynomial fit could solve this problem, but my linear regression model was only able to do mostly a simpler linear fit. We can see in the plot of the residuals that they are highly correlated with our predictor. This means our fit is really bad and we need to get a better one in order to make any kind of predictions.

Overall, both models are interesting for analyzing specific trends and have their own use cases, but are both pretty bad and need major improvements, which is what I’d expect given I’m trying to do a lot (predict fine precipitation values) with very little (coarse temperature data).

# Conclusion

This is a pretty complicated problem. It feels like the use case for each model should be considered carefully before choosing one that is appropriate for using on gridded data, such as whether you’re predicting one gridcell or a regional scale. Simple linear regression methods look like they’re not even close to good enough, while PCA-KNN approaches capture some information at a regional scale but are also often inaccurate. Overall, it seems like much, much work should be added for either of my models to make them better and more accurately represent the precipitation data.

Citations: I discussed general approaches with Lily, Maddie, Catharine, and Kyle. I used Chat-GPT and GitHub Co-Pilot to generate ideas on how to implement functions and solve errors, but all code here is my own or has been significantly modified to align with my specific methods.