# Project 1

## Precipitation Downscaling using PCA-KNN Combined Approach

Samira Hossain (sh162)

2023-11-13

## Table of contents

# 1. Executive Summary

The main objective of this project is the downscaling of precipitation data across Texas. This involves refining the spatial resolution of precipitation forecasts by using two critical regional climate variables as predictors- air temperature and dew point temperature. The choice of air temperature and dew point temperature as predictors is grounded in their significant meteorological relevance. Air Temperature is a fundamental climatic variable that directly influences evaporation, condensation, and the capacity of the air to hold moisture – all key elements in the precipitation process. Dew point temperature on the other hand is a measure of atmospheric moisture and is important in understanding the humidity and water vapor content in the air, which are directly related to precipitation formation.

The project focuses on developing a method to accurately downscale precipitation from these predictors, transitioning from their lower-resolution datasets to the higher-resolution observed precipitation data. Two models are employed: KNN-PCA Model: A sophisticated model combining K-Nearest Neighbors (KNN) and Principal Component Analysis (PCA) forms the core of our approach. This model combines KNN to model complex, non-linear relationships and PCA for its effectiveness in reducing data dimensionality, making it particularly suited for handling varied resolution scales. Mean-Based Prediction serves as a baseline method and provides a benchmark to evaluate the advanced KNN-PCA model.The KNN-PCA model demonstrates a marked improvement over the baseline mean-based prediction method. This is reflected in its higher accuracy and enhanced spatial resolution of precipitation forecasts.

This project outlines a framework for precipitation downscaling and also reveals some of the limitations. It underlines the importance of air temperature and dew point temperature as key predictors in precipitation modeling and establishes the KNN-PCA model as a useful tool in regional climate analysis. However, while it performs better than the very simple baseline method, it should be compared with other established methods in machine learning for further judgment on its efefctiveness.

## 1.1 Key summary of datasets

### 1.1.1 Region of interest: Texas

1.1.2 Predictors: Air Temperature and Dew Point Temperature

1.1.3 Source of Data: ERA5 Reanalysis

1.1.4 Predictand: Observed Precipitation over Texas

1.1.5 Source of Data: CPC

# 2. Exploratory Data Analysis

We begin with exploring the datasets. In this project, we are working with air temperature and dew point temperature as our inputs or predictors. As our y or predictand, we have observed precipitation data over Texas. The former are coarser data downloaded from ERA5 Reanalysis website using a julia script while the latter is a fine resolution data. To meet the project goal, we observe the temporal and spatial dimensions and subsets according to our requirements.

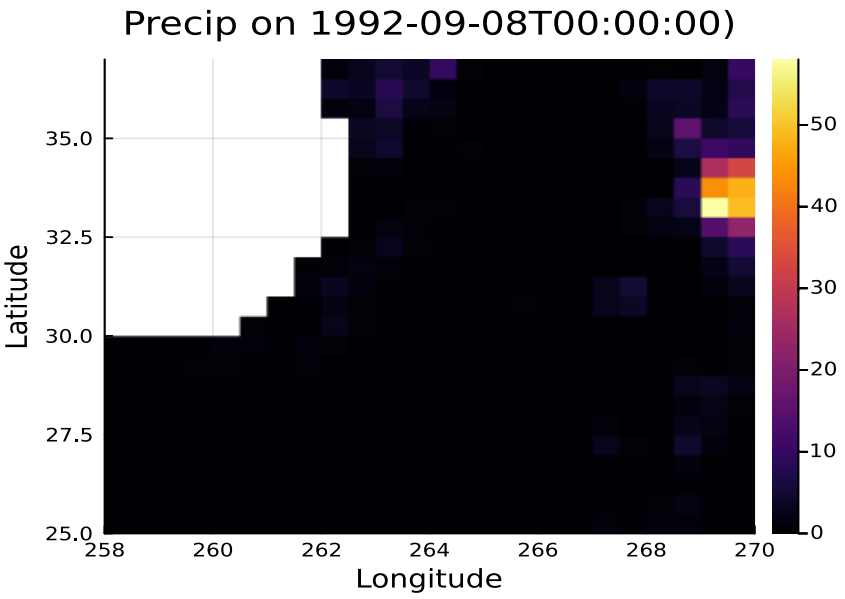
## 2.1. Checking dimensions and visualizing the data

We at first read in the finer resolution data, which is the precipitation data over Texas. As the output shows, it is observed precipitation data for Texas region. The spatial resolution is 24*24 for 1979 to 2020, thus could be called finer than the inputs of our model. After reading in the data, we reverse the latitudes and then plot a specific day's data to see how the data look

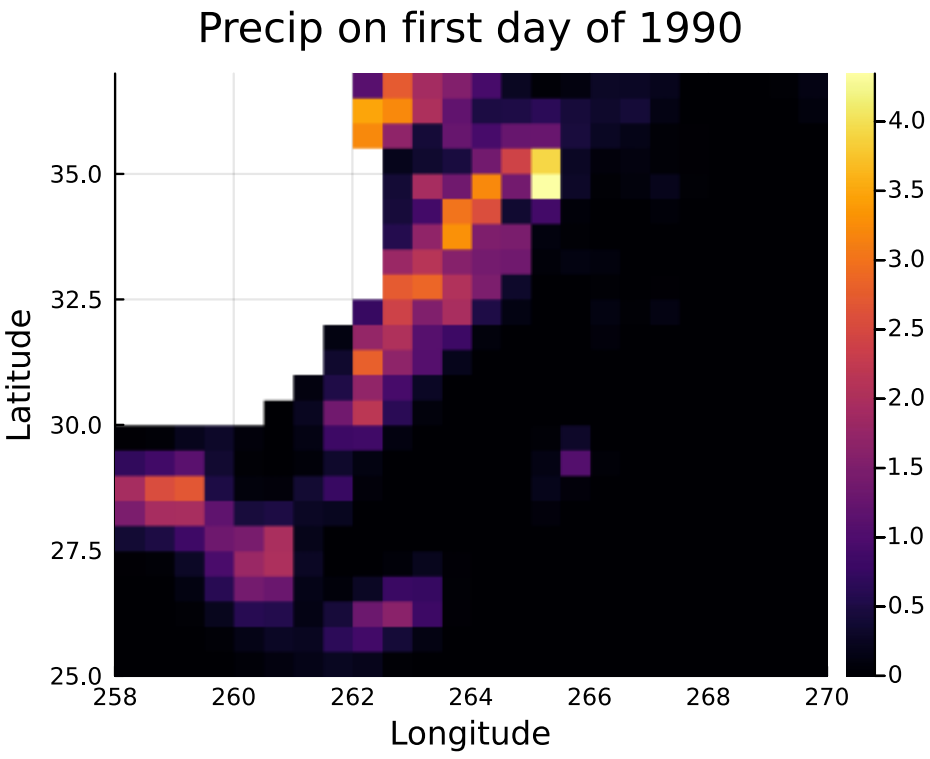
```
../data/precip_data_Houston/precip_tx.nc
Group:                                                                  /

Dimensions
  lat                                   =                              24
  lon                                   =                              24
  time                                  =                           16365

Variables
 lat                                                                  (24)
  Datatype:                   Union{Missing,          Float32}      (Float32)
  Dimensions:                                                          lat
```

```
  Attributes:
   _FillValue                  = NaN
   actual_range                = Float32[89.75, -89.75]
   long_name                   = Latitude
   units                       = degrees_north
   axis                        = Y
   standard_name               = latitude
   coordinate_defines          = center

lon                                                    (24)
  Datatype:    Union{Missing, Float32} (Float32)
  Dimensions:  lon
  Attributes:
   _FillValue                  = NaN
   long_name                   = Longitude
   units                       = degrees_east
   axis                        = X
   standard_name               = longitude
   actual_range                = Float32[0.25, 359.75]
   coordinate_defines          = center

time                                                   (16365)
  Datatype:    DateTime (Int64)
  Dimensions:  time
  Attributes:
   long_name                   = Time
   axis                        = T
   standard_name               = time
   coordinate_defines          = start
   delta_t                     = 0000-00-01 00:00:00
   avg_period                  = 0000-00-01 00:00:00
   units                       = days since 1979-01-01 00:00:00
   calendar                    = proleptic_gregorian
```

```
precip (24 × 24 × 16365)
  Datatype: Union{Missing, Float32} (Float32)
  Dimensions: lon × lat × time
  Attributes:
   _FillValue = NaN
   var_desc = Precipitation
   level_desc = Surface
   statistic = Total


24-element Vector{Union{Missing, Float32}}:
 36.75f0
 36.25f0

 .
 .
 26.25f0
 25.75f0
 25.25f0

24-element Vector{Union{Missing, Float32}}:
 258.25f0
 258.75f0
 ……
 269.25f0
 269.75f0
```




Since we are working with 31 years of data (11,323 days), we extracted the data of 1990 to 2020 from the given dataset. Plotting the data at random day, we can see how precipitation changes over time.



We also read in the data for the input variables in our model- which is air temperature and dew point temperature. We have used a given function called open_mfsataset to read in the data.The data have been downloaded from ERA5 and these are much coarse resolution data compared to our predictand or precipitation.

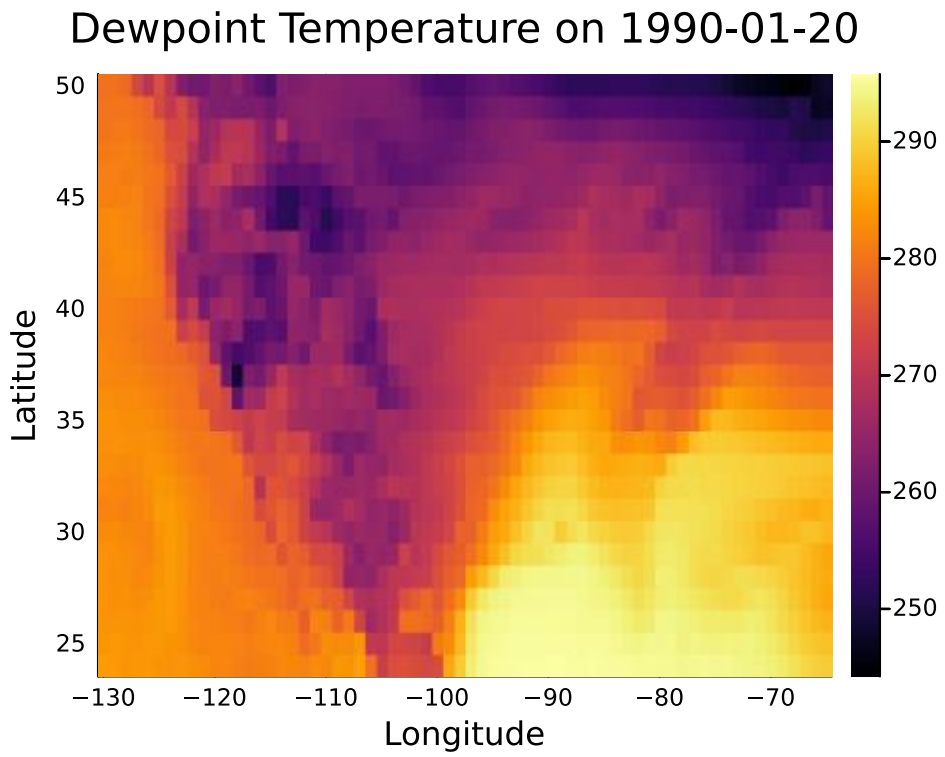
```
open_mfdataset (generic function with 1 method)
```

```
Dict{String, Vector} with 4 entries:
  "latitude"  => Float32[50.0, 49.0, 48.0, 47.0, 46.0, 45.0, 44.0, 43.0, 42.0, …
  "time"      => [DateTime("1990-01-01T00:00:00"), DateTime("1990-01-01T01:00:0…
  "longitude" => Float32[-130.0, -129.0, -128.0, -127.0, -126.0, -125.0, -124.0…
  "t2m"       => Union{Missing, Float64}[279.473, 280.155, 280.676, 278.855, 27…
```

As observed, the longitude and latitude of the ERA5 data are for the whole North America, we can visualize it in the plot as well. So, we subset the data to contain only Texas's air temperature and dew point temperature.

```
2046-element Vector{Float32}:
 -130.0
 -129.0
 -128.0
 -127.0
    ⋮
  -76.0
  -65.0
```

The data dictionary contains 31 years of hourly data for temperature and dew point. So, we converted the hourly data to daily data by taking average or mean. In order to do so, we first reshape the aggregrated data in typical three dimensional structure as netcdf datatypes. After reshaping, we convert the hourly data to daily data.




This is the original ERA5 dew temperature data for the North America. As seen from the plot, there are clear geographic contributions can be seen for air temperature which is interesting.

# 2. Methodology

After preparing the datasets for our specific regional and temporal interest in the previous step, the KNN-PCA method is now applied. At first, we applied some data processing steps and did principal component analysis to observe its effect in our feature space. Then, we combined them with the KNN method.

## 2.1. Splitting the data to training and testing sets

After exploring the data and doing subsetting over time and location, we now split the data for training and testing. We considered daily air temperature and dew point temperature data from 1990 to 2010 to be training data and data from 2011 to 2020 to be testing data. We can see the dew point data split into training and testing as an example, in the output.

## 2.2. Preprocessing of data for PCA

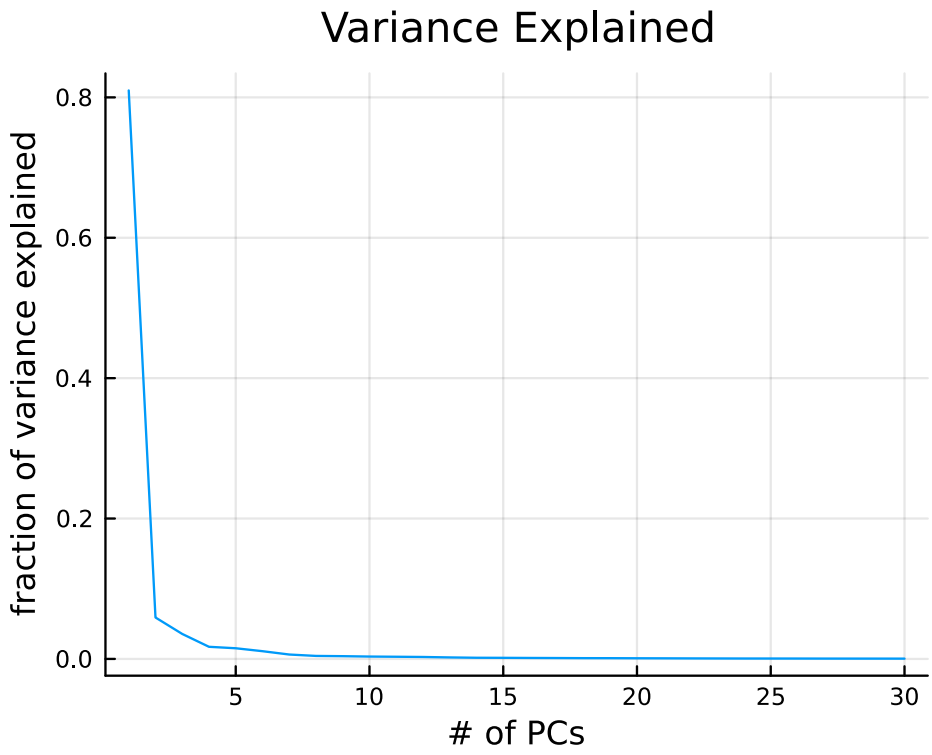
Before doing PCA, we first preprocess our input datasets for air temperature and dewpoint by normalizing and rescaling them.The 3D datasets are also converted to 2D structure before doing PCA. Applying the preprocess function over the training and testing datasets, we could see in the output for one of the datasets that it has now transformed to a 2D rescaled data.

```
192×7671                                                    Matrix{Float64}:
 -0.967929   -0.751383   -0.569463   …   -0.878923   -0.307995   -0.698149
 -1.10327    -1.12949    -0.492027       -0.729157   -0.708934   -1.00943
 -0.634328   -1.14089    -0.216078       -1.02411    -1.16804    -1.2419
  ⋮                                   ⋱                           ⋮
 -1.70423    -1.07315    -0.262314       -0.614956   -1.17133    -2.44951
 -1.7226     -1.18874    -0.280972       -0.160799   -0.864181   -2.47125
 -1.71366    -1.35204    -0.351218        0.271067   -0.481142   -2.38365
 -1.62577    -1.48022    -0.471277        0.295442   -0.196807   -2.28633
 -1.61352    -1.56444    -0.459289   …    0.262395    0.0447962  -2.20979
------
```

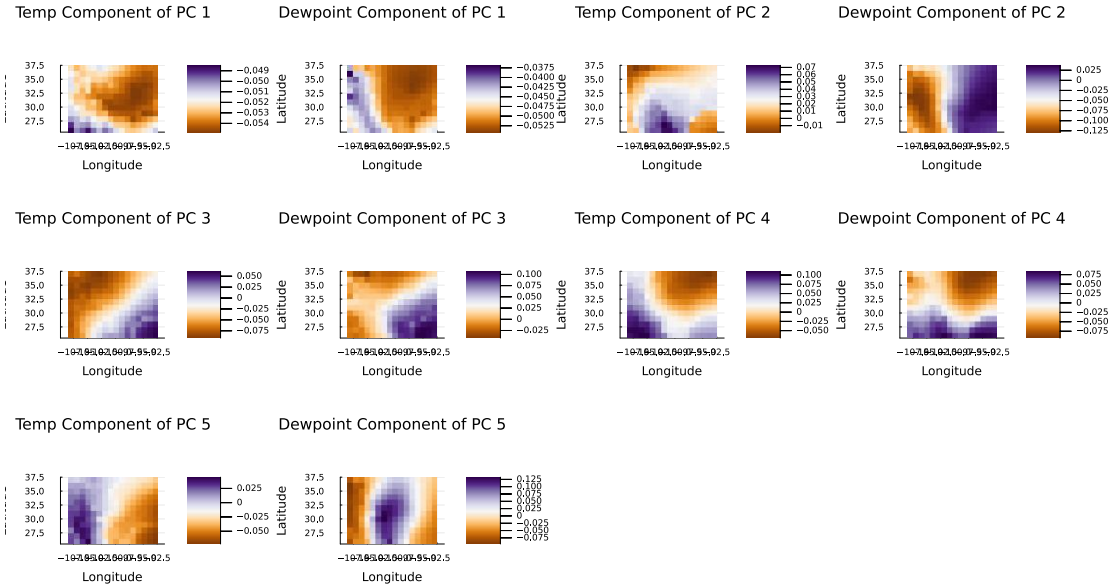
Because we have two variables- air temperature and dew point temperature, before doing PCA, we concate them to a single matrix. This concatenated dataset is our X(input) on which PCA will be done.

From the variance plots, we could observe that the first 5 principal components seem to capture most of the variance in the feature space.

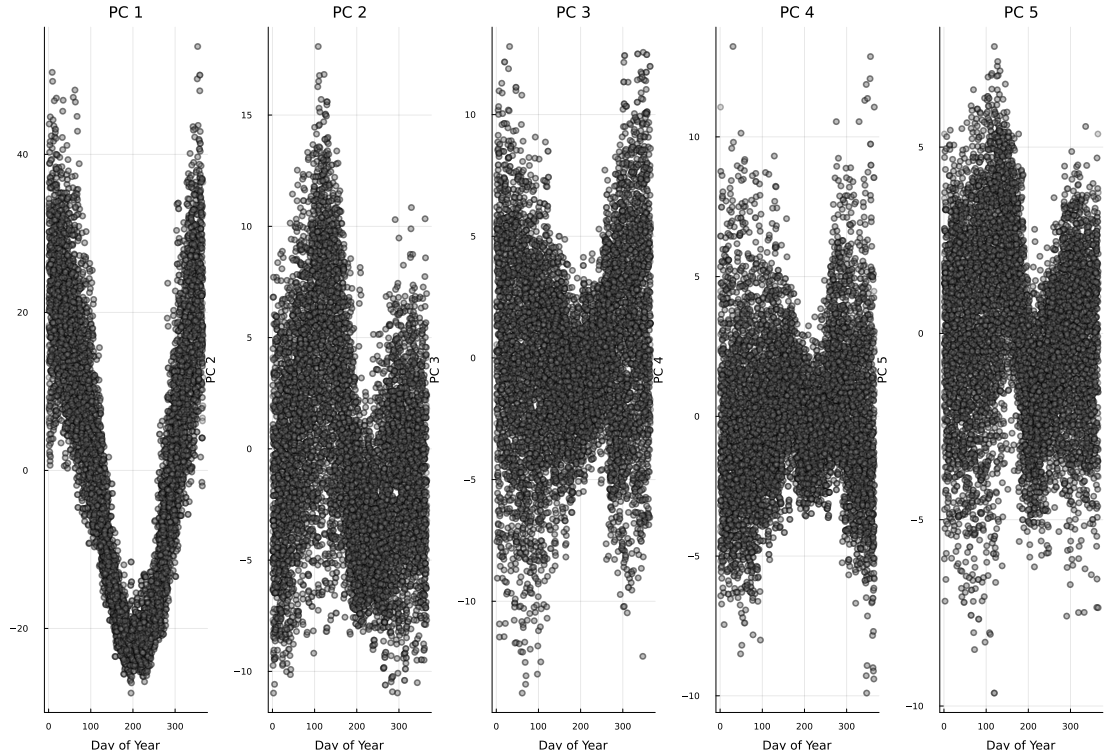
```
30-element Vector{Float64}:
 311.0148477856334
  22.66721814656555
  13.667937327161921
   ⋮
   0.16446888601568477
   0.15428803374324543
```



We can plot the principal components for both air temperature and dewpoint temperature and observe the plots. It seems the first pc as usual captures the most variance for both air temperature and dew point temperature.



The scatter plot reconfirms this as it shows that the first pc capturing a large variance, showing how seasonal changes happen, while the other pcs do not vary as much and might just accounting for small variations.



## 2.3. K Nearest Neighbor Analysis

After the dimensionality reduction, we then apply resampling based KNN method for prediction or downscaling precipitation. We write a knn function to implement the algorithm, which is explained as below:

1. Choosing the Number of Neighbors, K

We decide how many neighbors (K) will contribute to determining the output for a new data point. The choice of K is a hyperparameter, and we tuned it to get the best result.A smaller K makes the algorithm sensitive to noise in the data, a larger K makes it computationally difficult so we optimize a suitable K to balance it.

2. Calculating Distance

For a given new data point, we calculate the distance between this point and all other points in the training dataset.In this case, we are using euclidian distance because it is commonly used but there are other options too. This step helps us to identify which points in the training data are ‘nearest’ to the new data point.

3.Identifying Nearest Neighbors

We identify the K nearest points (neighbors) to the new data point based on the calculated distances.These nearest neighbors will ‘vote’ or contribute to predicting the output for the new data point.

4. Aggregating Neighbor Information

The approach we used here is a regression based.We calculate the weighted average of the response values of the K nearest neighbors.The average value provides the best estimate for the continuous output of the new data point.

5. Making prediction

The aggregation from the previous step is used as the predicted output for the new data point.This step finalizes the prediction process by using the 'wisdom of the crowd' from the nearest neighbors.We implement these steps by a function called knn, which is embedded with functions for calculating euclidian distance, finding nearest nneighbour etc.

Since the pca analysis seems to give desired results, we now combine pca with knn and made a function called predict_knn_combined, which takes the train and test datasets, does dimensionality reductions, finds nearest neighbours and predicts precipitation.

```
predict_knn_combined (generic function with 1 method)
```

## 2.4.Hyperparameter Tuning

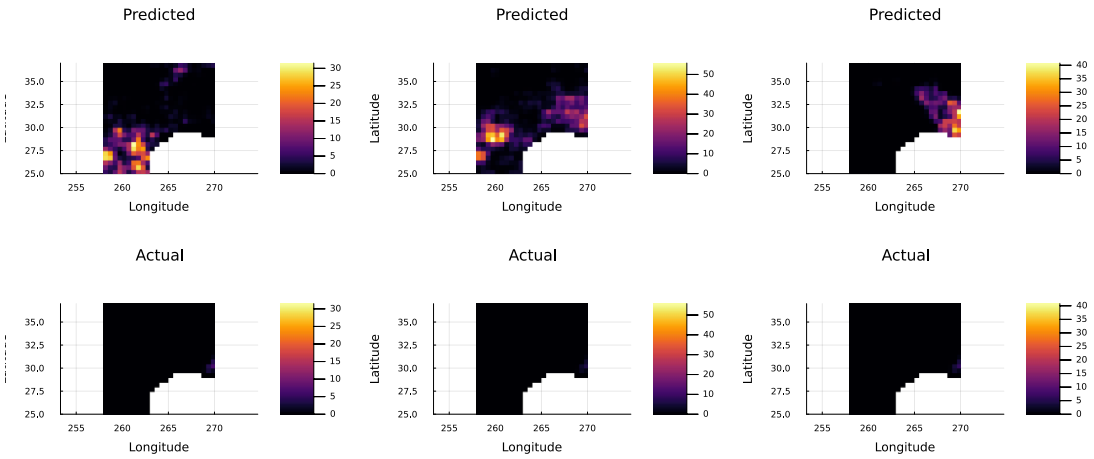
In this case, we had two hyperparameters to play around with-number of principal component (n_pca) and nearest neighbour value K. we found that keeping K=3 and n_pca =10 gives the lowest mean squared error (mse) result.

## 3. Results

We validate our precipitation prediction by plotting it in comparison with the actual given data of precipitation. We select three random time points from test data for precipitation and predict precipitation. Visualizing the predicted values, we could observe the the model was able to predict precipitation in various locations compared to actual precipitation.

```
3-element Vector{Matrix{Union{Missing, Float32}}}:
 [5.1847515f0 9.306382f0 … 0.052618943f0 0.0f0; 3.4392428f0 4.5242987f0 … 0.050748385f0 0.0f0; … ; missing missing … 0.04371802f0 2.7516046f0; missing missing … 0.032738116f0 3.898173f0]
 [1.0564678f0 1.9646056f0 … 0.0f0 0.0f0; 3.8374887f0 3.4213696f0 … 0.0f0 0.0f0; … ; missing missing … 0.594733f0 0.70078754f0; missing missing … 3.4787986f0 1.1400607f0]
```

[0.0f0 0.0f0 … 0.0f0 0.0f0; 0.0f0 0.0f0 … 0.0f0 0.0f0; … ; missing missing … 0.0f0 0.0f0; missing missing … 0.0f0 0.0f0]




The heatmaps showing the predicted vs the actual is a way to validate the model. As we can see, the predicted precipitation is indicating precipitation values at latitudes and longitudes where previously no data was observed. Also, the data seems to suggest there is higher precipitation values at the southeast. While these heatmaps are not exactly clear enough to see the geographical attributes, it is possible to identify and relate it to it. We can try and apply this model to individual variables and see how they predict precipitation, however in this project we are considering both variables.

## 3.1. Comparison with a baseline method

As a baseline method, a very simple method called Mean Prediction Method.In this method, the average (mean) value of the target variable from the training dataset is used as a prediction for all instances in the test dataset. It assumes that the best guess for any future value is the average of the past observed values. We find that knn-pca method outperforms the baseline method by comparing their mean squared errors. However, the difference is not staggering, hence the KNN PCA model seems to be not as efficient.

67.77051f0 for KNN-PCA

71.716675f0 for baseline method

### 3.2. Limitations of the Results

Several limitations exist that are likely impacting the results. Firstly, the hyperparameter tuning is not done efficiently as it is done manually changed k and n_pca. A function which takes in several K values and n_pca values and return a comparison may have showed how the changed are affecting the results. Secondly, the baseline method is too simple and it is usually outperformed by most modern methods, hence choosing a slightly more complex method would have given a better understanding of the performance of knn.

## 4. Conclusion

KNN combined with principal component analysis for downscaling precipitation over Texas based on predictors air temperature and dew point temperature give satisfactory results compared with simple baseline method such as Mean Prediction Method. The hyperparameter of K=3 and number of principal component being 10 yielded the lowest mean squared error. The downscaling was verified in several visualization compared with the actual precipitation data.The KNN-PCA model has some challenged as it requires rigorous data preprocessing steps and is computationally challenging when working with lage datasets, however it is straightforward as a method.