Comparative Climatic Analysis: Understanding the World Around Us

Brian Dong
Julian Duran
Skyler Gossman
Pedram Mirshahreza

Introduction

The National Oceanic and Atmospheric Administration's (NOAA) mission is to understand and predict changes in climate, weather, ocean and coasts; they share this knowledge with others in the effort to conserve and manage coastal and marine ecosystems and resources. The NOAA is an organization that provides products and services to support economic vitality. Their scientists use cutting edge research and instruments to provide citizens, planners, emergency managers and other decision makers with reliable information. Their mission is to better understand the natural world, and to help protect resources through the monitorization of global weather and climate. Further, this agency holds leadership roles in shaping international ocean, fisheries, climate, space and weather policies. They work closely with other nations in order to predict and respond to changes in climate and other environmental challenges.

It is important to collect weather data because it helps with prediction. These predictions can be as simple as the meteorologists or storm chasers who use them for weather predictions we see on the news, or they can also be used for more in-depth analysis in understanding global warming or other important ecological topics. The NOAA utilizes National Weather Service (NWS) sites located throughout the entire United States as well as its territories. They provide annual reports and maintain high transparency with their data. One such report, like the one used in this analysis, is the comparative climatic database. Although these specific data tables aren't meant for the analysis of climate variability and change, they can be used for reference and other, non-specific applications such as comparing general climate conditions of measures such as temperature, precipitation, winds, etc. As such, they have a large number of files and tables which can be used by the curiously minded. In fact, they are formatted more as reference tables, less for data analysis; this facet will be discussed later as a major limitation to this study.

With this data, the main goal is to withdraw from the NOAA's vast compendium, organize, then finally interpret the data. It is the aim of this study to simply understand differences between regions of the United States, and also gain a better understanding of relationships between and within them. With this newly-found understanding, further studies can be implemented which dives deeper into climatic data, its trends, and quite possibly predictive endeavors can be undertaken.

Namely, the relationship between average snowfall and other variables in the dataset was explored. From this, a predictive model for snowfall and temperature was created as they have a strong relationship. Strong relationships were also shown to exist between snowfall and possible sunshine (which is expected). However, relationships between other climatological variables and average snowfall were not as clear; factors like humidity may have a significant effect on changes in snowfall as well.

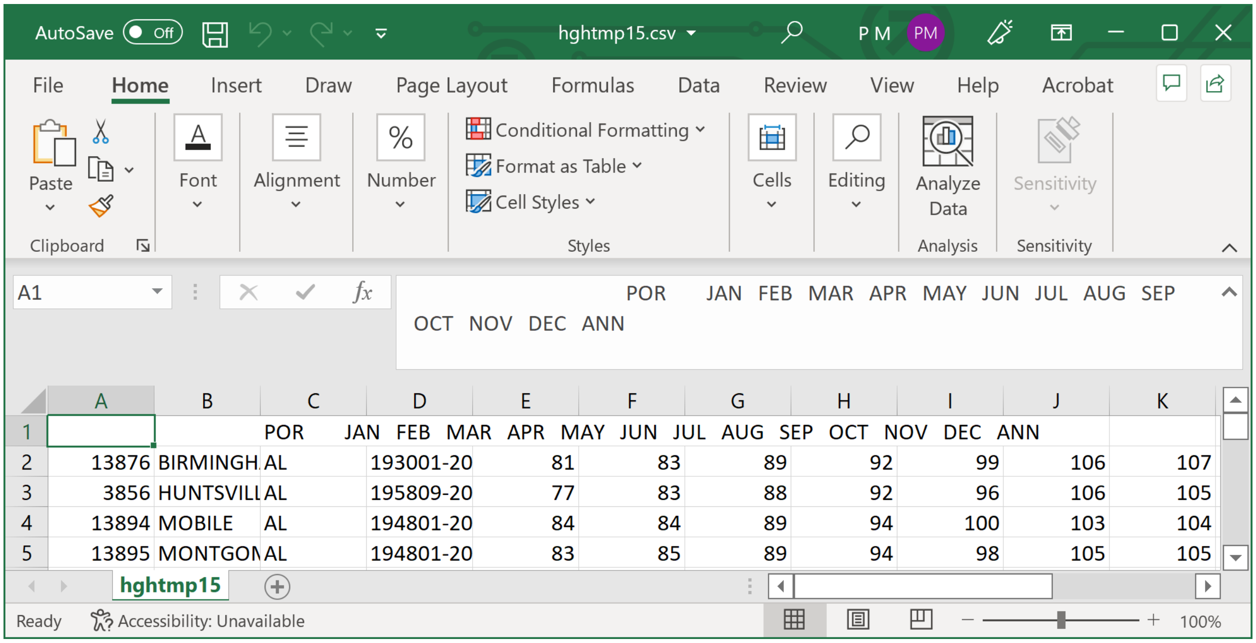
Another aspect of the dataset we explored was the relationship between the climatic region of the state that the weather station resides in, and its climatic measurements. The many regions of the United States experience different climatic conditions, and therefore have different measurements of average snowfall, humidity, etc. We explored both the general differences in climatic conditions between regions, as well as a specific focus on stations in Hawaii and Alaska. Due to the geographical distance between the stations in the continental United States and the non-continental stations in Hawaii and Alaska, it is difficult to place the stations in these two states into a definite region similar to stations on the continental United States. We want to explore the similarities between stations in Hawaii and Alaska, and stations in the continental United States, and find which regions have the most similar climatic conditions to the stations in these two states.

Methods

The data presented was composed of numerous individual files of various readings from across the United States; there was a file each for more than approximately 130 cities. In each file, there were measures of temperature by month, snowfall and rainfall amounts, days of sunshine and cloudiness, and humidity. Using python and numerous techniques were employed to extract pertinent data points from each file, then a master file was created with the averages over all respective measures. US States were then further divided into regions for comparison. Following the data reorganization, an Exploratory Data Analysis was completed to become familiar with the variables at hand. It is important to note the resulting variables represent the *averages* of each specific city. Although this does induce limitations to this study, it was a necessary undertaking in order to effectively analyze these data in a respectable amount of time.

Data Cleaning, Data Preprocessing and Creating New Datasets

The original set of data which were provided were stored in numerous CSV files. Each file were named after each attribute (measured climate variables) for 135 to 277 cities in the U.S. Some of these 32 files were associated to the same attributes but for differend dates of updates. At this point, the number and type of distinct attributes were determined and from the set of prvided files, the most updated ones were selected.

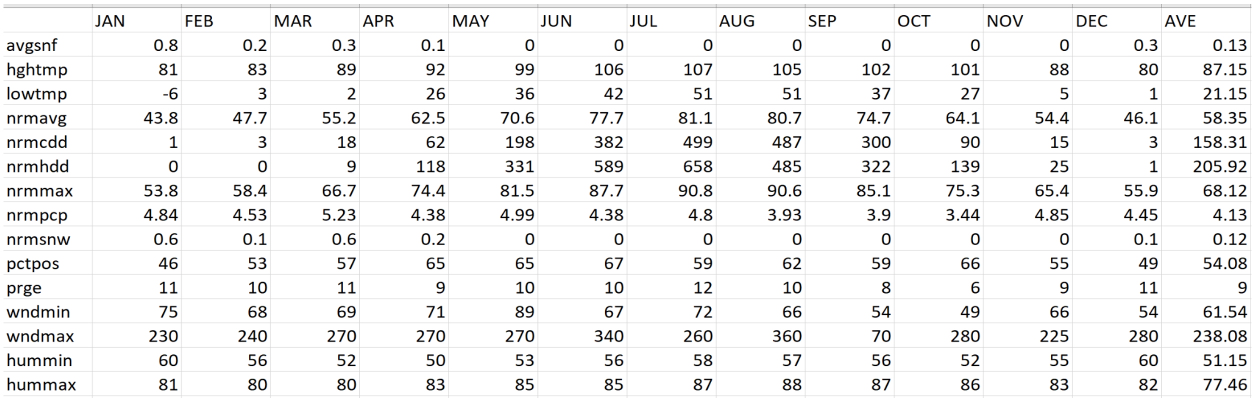


POR JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC ANN

	A	B	C	D	E	F	G	H	I	J	K
1	POR JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC ANN										
2	13876	BIRMINGH	AL	193001-20	81	83	89	92	99	106	107
3	3856	HUNTSVILL	AL	195809-20	77	83	88	92	96	106	105
4	13894	MOBILE	AL	194801-20	84	84	89	94	100	103	104
5	13895	MONTGON	AL	194801-20	83	85	89	94	98	105	105

Figure 1. An example of the original files provided for the project. The headers are unappporiately set and the indexings were not suitable to work with

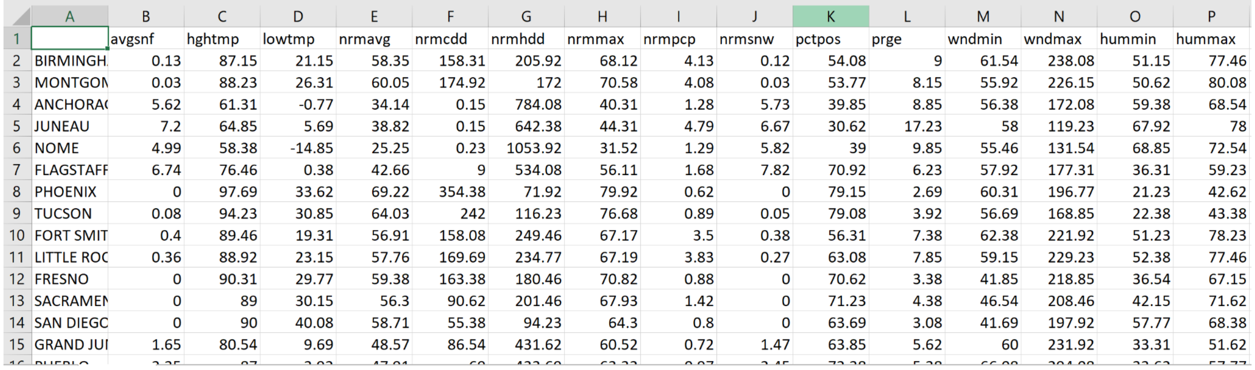
In order to creat a major dataset, from the newly selected 15 files, the file with smallest set of the data points were selected. Then, utilizing Python, each data points were selected and set as a reference. After that part, the similar cities and associated data points were extracted from the other 14 newly selected files and all were stored in csv files, named by the cities and containing the measurements of each attribute.



	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	AVE
avgsnf	0.8	0.2	0.3	0.1	0	0	0	0	0	0	0	0.3	0.13
hghtmp	81	83	89	92	99	106	107	105	102	101	88	80	87.15
lowtmp	-6	3	2	26	36	42	51	51	37	27	5	1	21.15
nrmavg	43.8	47.7	55.2	62.5	70.6	77.7	81.1	80.7	74.7	64.1	54.4	46.1	58.35
nrmcdd	1	3	18	62	198	382	499	487	300	90	15	3	158.31
nrmhdd	0	0	9	118	331	589	658	485	322	139	25	1	205.92
nrmmax	53.8	58.4	66.7	74.4	81.5	87.7	90.8	90.6	85.1	75.3	65.4	55.9	68.12
nrmpcp	4.84	4.53	5.23	4.38	4.99	4.38	4.8	3.93	3.9	3.44	4.85	4.45	4.13
nrmsnw	0.6	0.1	0.6	0.2	0	0	0	0	0	0	0	0.1	0.12
pctpos	46	53	57	65	65	67	59	62	59	66	55	49	54.08
prge	11	10	11	9	10	10	12	10	8	6	9	11	9
wndmin	75	68	69	71	89	67	72	66	54	49	66	54	61.54
wndmax	230	240	270	270	270	340	260	360	70	280	225	280	238.08
hummin	60	56	52	50	53	56	58	57	56	52	55	60	51.15
hummax	81	80	80	83	85	85	87	88	87	86	83	82	77.46

Figure 2. There were 134 files generated for each city, rows representing the attributes and the columns the average of each month followed by the average of 12 months.

In this process all of the nonnumerical entries, empty entries and the ones with obviously wrong data were deleted. All of the values of the average column of each city.csv file were then stored in a major dataset called main.csv. This data set contained the rows for 134 cities in which the average value of attributes were stored in their columns.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		avgsnf	hghtmp	lowtmp	nrmavg	nrmcdd	nrmhdd	nrmmax	nrmpcp	nrmsnw	pctpos	prge	wndmin	wndmax	hummin	hummax
2	BIRMINGH.	0.13	87.15	21.15	58.35	158.31	205.92	68.12	4.13	0.12	54.08	9	61.54	238.08	51.15	77.46
3	MONTGON	0.03	88.23	26.31	60.05	174.92	172	70.58	4.08	0.03	53.77	8.15	55.92	226.15	50.62	80.08
4	ANCHORA(	5.62	61.31	-0.77	34.14	0.15	784.08	40.31	1.28	5.73	39.85	8.85	56.38	172.08	59.38	68.54
5	JUNEAU	7.2	64.85	5.69	38.82	0.15	642.38	44.31	4.79	6.67	30.62	17.23	58	119.23	67.92	78
6	NOME	4.99	58.38	-14.85	25.25	0.23	1053.92	31.52	1.29	5.82	39	9.85	55.46	131.54	68.85	72.54
7	FLAGSTAFF	6.74	76.46	0.38	42.66	9	534.08	56.11	1.68	7.82	70.92	6.23	57.92	177.31	36.31	59.23
8	PHOENIX	0	97.69	33.62	69.22	354.38	71.92	79.92	0.62	0	79.15	2.69	60.31	196.77	21.23	42.62
9	TUCSON	0.08	94.23	30.85	64.03	242	116.23	76.68	0.89	0.05	79.08	3.92	56.69	168.85	22.38	43.38
10	FORT SMIT	0.4	89.46	19.31	56.91	158.08	249.46	67.17	3.5	0.38	56.31	7.38	62.38	221.92	51.23	78.23
11	LITTLE ROC	0.36	88.92	23.15	57.76	169.69	234.77	67.19	3.83	0.27	63.08	7.85	59.15	229.23	52.38	77.46
12	FRESNO	0	90.31	29.77	59.38	163.38	180.46	70.82	0.88	0	70.62	3.38	41.85	218.85	36.54	67.15
13	SACRAMEN	0	89	30.15	56.3	90.62	201.46	67.93	1.42	0	71.23	4.38	46.54	208.46	42.15	71.62
14	SAN DIEGC	0	90	40.08	58.71	55.38	94.23	64.3	0.8	0	63.69	3.08	41.69	197.92	57.77	68.38
15	GRAND JUI	1.65	80.54	9.69	48.57	86.54	431.62	60.52	0.72	1.47	63.85	5.62	60	231.92	33.31	51.62

Figure 3. The main dataset: each row is associated to one city in the U.S. and each column represent an attribute.

Average Snowfall

Since the dataset contains both observed values and climatological normals, we focused on modeling average snowfall using the variables of the observed values. For the purpose of modeling average snowfall, we removed the variables that contained climatological normals in order to keep the data collection methods of the analyzed variables consistent. The variables considered for inclusion in the model are: high temperature, low temperature, mean days with greater than 0.01 inches of precipitation, percentage of possible sunshine, both minimum and maximum humidity, and both minimum and maximum wind speed.

Average snowfall among the weather stations is highly skewed to the right, suggesting that the majority of weather stations experience little snowfall on average. To analyze the relationship between average snowfall and other climatic measurements, we first constructed a linear regression model using average snowfall as the dependent variable. Model diagnostics for our linear regression model on snowfall showed many problems. Checking model assumptions, the residual plot had a clear fanning pattern, and the normal quantile plot also saw large deviations from a normal distribution at the tails. It is clear that our initial model does not meet the assumptions of homoscedasticity and normality. We considered log transformations and a Box-Cox transformation, but after checking model diagnostics the assumptions remained violated. The Box-Cox transformed model was normally distributed, but did not have constant variance. In these transformed models, we found that there were multicollinearity problems from highest record temperature and minimum humidity, which are both highly related to their respective counterpart variables. We paid attention to these two variables in later models of average snowfall. Because of these violations in assumptions, we decided against modeling average snowfall using this method of linear regression.

To continue modeling average snowfall and its non-normal distribution, we considered generalized linear models, and checked different distribution families to find an appropriate fit for our variable. Because average snowfall is a positive-skewed continuous variable, we considered a model under a gamma distribution family using a log link. We also considered average snowfall as a binary variable, as there may be significant differences between stations that receive little to no snowfall, and stations that receive higher average snowfall. To model this, we performed a logistic regression analysis using a generalized linear model with a logit link. Because of the different assumptions of these models, we decided to test their predictive ability by splitting the dataset into a training and test set, and using the models of the training set to check the predictive accuracy for observations of the test set.

Our initial fit of the generalized linear model with a gamma family for the dependent variable showed similar issues with multicollinearity. Minimum humidity again had a large VIF of over 7, and we decided to remove the variable from the model. We performed backward variable selection on the model until reaching a final model with significant covariates. Residual plots of the fitted model show a very clear pattern in the residuals, suggesting that there may still be issues regarding the assumptions of our model.

For our logistic regression using average snowfall, the variable was divided into a factor variable with two levels: stations that received greater than 0.01 inches of snowfall on average, and stations that received that amount of snowfall or less on average. This cutoff point was decided based on the same measurement as the variable that measures precipitation in this dataset, which defines its cutoff point for mean number of days with precipitation as 0.01 inches.

Our initial logistic regression model was unable to converge, and after looking at some diagnostics, we found that there was a severe multicollinearity problem amongst the predictor variables. We experienced difficulty in performing an appropriate model selection process for our logistic model, as all of our covariates were not statistically significant, and multicollinearity problems continued to persist even after removal of variables with very large VIF values like minimum humidity. We therefore decided to ignore instances of multicollinearity for the purposes of this model, and instead use AIC as the criterion for model selection. This selection process led to a final model with average percent of possible sunshine, maximum wind speed, and record low temperature. These covariates were not significant in the final model, and the associated residual plot showed a very clear pattern in the residuals. The diagnostics show us that a logistic regression model for average snowfall is not appropriate for an adequate analysis of the variable.

Regional Differences

To explore the relationship between climatic region and climatic measurements, we looked at the same variables as our analysis of snowfall: average snowfall, high temperature record, low temperature record, mean days with greater than 0.01 inches of precipitation, percentage of possible sunshine, both minimum and maximum humidity, and both minimum and maximum wind speed. We grouped each of the continental US states by the climatic region it resides in. In total, we specified seven climatic regions: Pacific, Rocky Mountain, Southwest, Midwest, Southeast, and Northeast. A final region termed Non-continental was created to include stations in Hawaii and Alaska. To assess the climate differences between regions, we considered a k-Nearest Neighbors model. To simplify the number of variables for the kNN algorithm, we averaged the respective low and high temperature, wind speed, and humidity variables. We then normalized each of the variables for the algorithm. We split the data randomly into a training set and test set, and then fit the kNN model to assess whether this classification by region is adequate in explaining differences in climatic conditions. We compared misclassification rates between choices of k from one to 30 to find the most accurate kNN model, which was at $k = 9$. We also fit a separate kNN algorithm to classify the Non-continental region stations in Hawaii and Alaska. For this algorithm, stations in the two non-continental states were used as the test set, with the rest of the regions used as the training set. To find the closest classification of the non-continental stations, we used the most accurate choice of k from the previous model, which was $k = 9$.

Exploratory Data Analysis

After the main dataset had been cleaned and reorganized, it consisted of 137 observations (one each per city), and 15 covariates for each given city. Table 1.0 provides descriptions and definitions for these covariates. Although not all covariates were used for our analysis, they are referenced in future sections of this report. Table 1.1 provides simple summary statistics for a subset

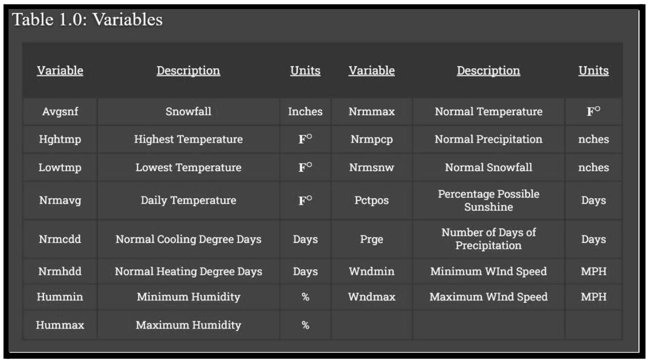


Table 1.0: Variables

Variable	Description	Units	Variable	Description	Units
Avgsnf	Snowfall	Inches	Nrmmax	Normal Temperature	F°
Hghtmp	Highest Temperature	F°	Nrmpcp	Normal Precipitation	nches
Lowtmp	Lowest Temperature	F°	Nrmsnw	Normal Snowfall	nches
Nrmavg	Daily Temperature	F°	Pctpos	Percentage Possible Sunshine	Days
Nrmcdd	Normal Cooling Degree Days	Days	Prge	Number of Days of Precipitation	Days
Nrmhdd	Normal Heating Degree Days	Days	Wndmin	Minimum WInd Speed	MPH
Hummin	Minimum Humidity	%	Wndmax	Maximum WInd Speed	MPH
Hummax	Maximum Humidity	%			

of the covariates presented in table 1.0; these values represent the overall average taken over all measures from the original dataset. For example, lowtmp (mean=15.2 degrees) represents the average low-temperature of all the observations across all cities nationwide. This averaging technique proves later to be a major self-induced limitation of this report, and will be discussed later in further sections.

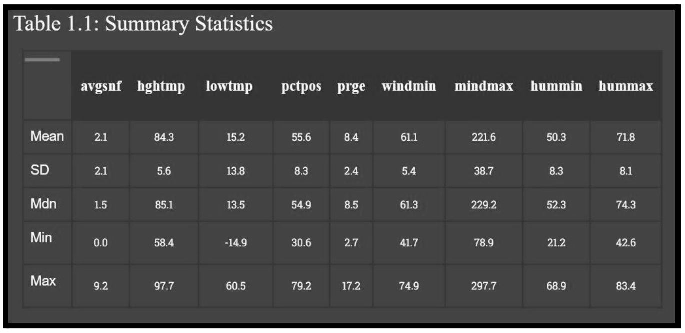
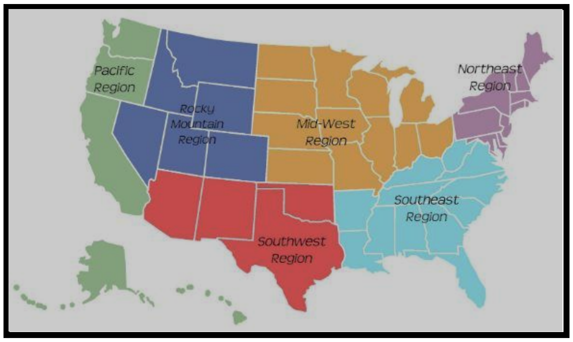


Table 1.1: Summary Statistics

	avgsnf	hghtmp	lowtmp	pctpos	prge	windmin	mindmax	hummin	hummax
Mean	2.1	84.3	15.2	55.6	8.4	61.1	221.6	50.3	71.8
SD	2.1	5.6	13.8	8.3	2.4	5.4	38.7	8.3	8.1
Mdn	1.5	85.1	13.5	54.9	8.5	61.3	229.2	52.3	74.3
Min	0.0	58.4	-14.9	30.6	2.7	41.7	78.9	21.2	42.6
Max	9.2	97.7	60.5	79.2	17.2	74.9	297.7	68.9	83.4



For portions of the analysis, cities across the United states were subset into 7 contiguous regions (US map). These regions are Pacific region (PC), Rocky Mountain (RM), Midwest (MW), Southwest (SW), Southeast (SE), and Northeast region (NE). Of these regions and cities, the highest average continuous rainfall was measured in Juneau, Alaska (17.2 days). Phoenix Arizona had the lowest average humidity (21.2%). Cities in Alaska and Hawaii were subset into a Non-continental (NC) region. Although these two US States differ vastly in terms of their geographical location and climate, they were organized in this way simply as a result of their being non-contiguous to the United States; the effect of this can be witnessed in following figures.

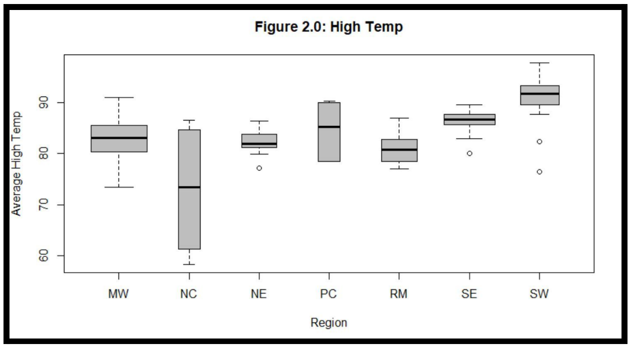


Figure 2.0: High Temp

When comparing the different regions with respect to average high temperatures (figure 2.0), a drastic difference can be seen in the SW region when compared to other regions; it is considerably higher. As such, average precipitation for the SW region (Figure 2.1) is drastically lower. The same cannot be said for other regions; although the differences are present, they are not as extreme as they are with the SW region. PC region precipitation amount varies wildly, second only to NC; this region represents a vast and varying climate region spanning California to Washington. Its average high temperature, however, is not necessarily higher than other regions, but it does vary less with a

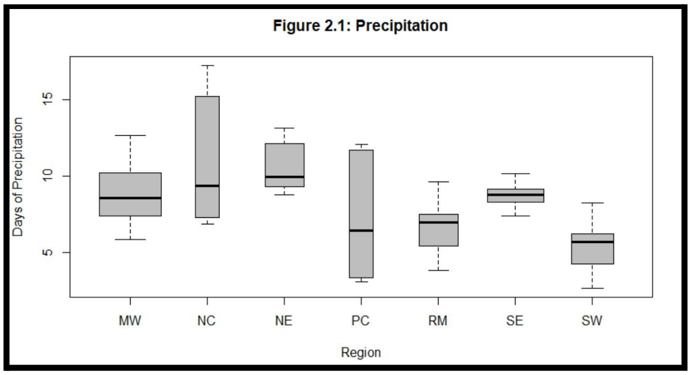


Figure 2.1: Precipitation

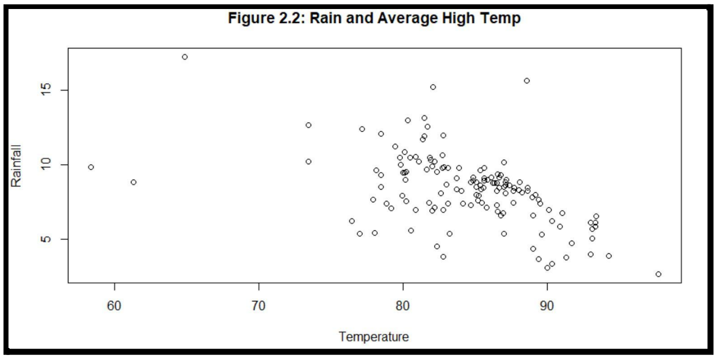


Figure 2.2: Rain and Average High Temp

slightly lower temperature median value; this could be a result of the state of Washington’s effect on the distribution of this measure, especially during the winter months. To verify these results, (Figure 2.2) shows the relationship between high temperature and rainfall; as temperature increases, rainfall decreases, on average. This trend appears to be more dramatic as temperature averages higher than roughly 85 degrees. As temperatures decrease, there appears to be more variability in average precipitation amounts, especially around 80 degrees. It is unknown whether or not this trend holds true for values less than 80 degrees though. Further, Pearson Correlation Coefficients (Figure 2.3) confirms and quantifies the negative correlation between average high temperature and precipitation. Similar results were observed when checking the Spearman Coefficients in the case of a non-linear relationship between the two (and all other) covariates.

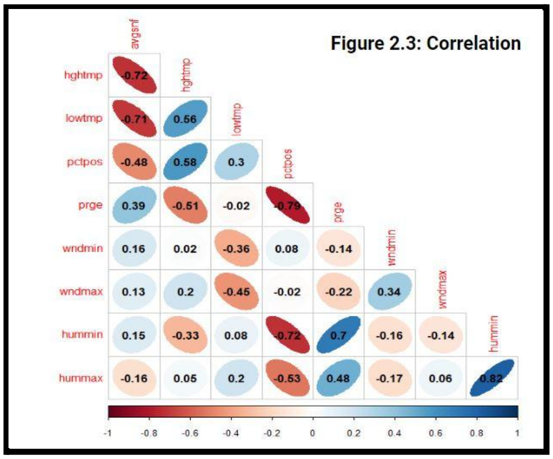
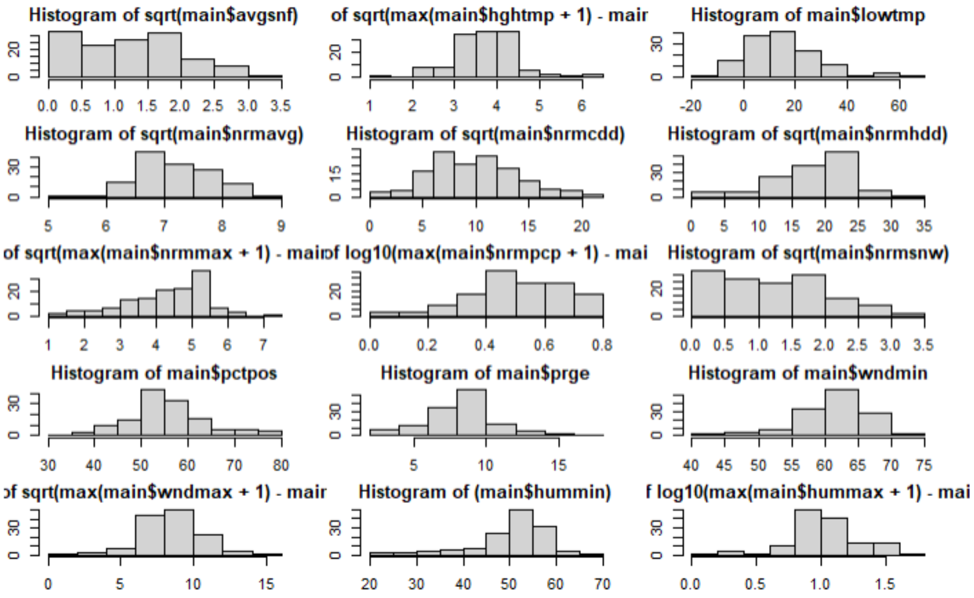


Figure 2.3: Correlation

Normalization

For the first part of our analysis methods, we utilize linear regression on all of the 15 weather attributes that we found averages for. Linear regression itself assumes normality since we have to derive the probability of the OLS estimators. For this reason we have to transform our data using a normalization method. From the figures below we have histograms of our attributes transformed primarily through square root and natural logarithmic. The proportion of Y to X squared follows that this method would work and the below figures shows the transformed data as histograms. Furthermore, following a method that pushes toward the central tendency involves making sure that there is an absence of multicollinearity which is an important assumption for factorial analysis since all factors are unique and orthogonal.. While it is true that Factorial Analysis does not require normality and can follow a moderately skewed distribution, there are still unique factors such as regression errors that assume normality.



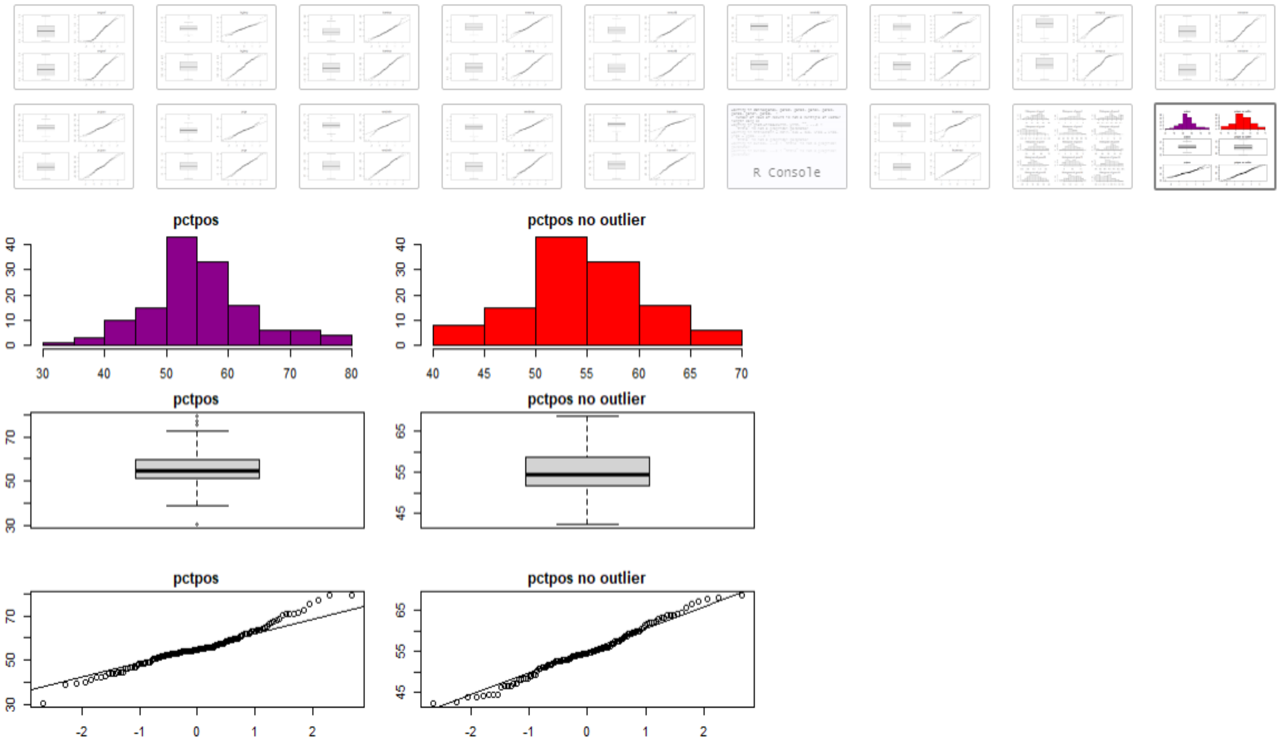
Dealing with outliers

Two of our other methods we utilize in this study PCA (principal component analysis) and Factorial Analysis follow non robust methods, and because of this it is important that we deal with outliers. If we examine our other non robust method PCA, we can understand it is a low rank decomposition of the data that minimizes the sum of the quadratic norms. In the bottom figure Y is the data and X is the PCA basis. Since PCA minimizes the quadratic norms it has the same issues as least squares and gaussian being sensitive to outliers. Since the outliers could drastically influence the PCA, they have to be dealt with.

$$\|Y - XA\|_F^2 = \sum_{j=1}^{m} \|Y_j - XA_{j.}\|^2$$

It is important to remember that in most datasets we should not recklessly remove outliers unless we are justified to do so. Doing so could skew the results. In this case most of the data for each of our attributes do not have outliers but some of them do, such as the percentage of possible sunshine. Normally there should not be dramatic outliers for weather data sets with a temperature reading of 107 Fahrenheit degrees when the highest average only reaches the 90's.

In most cases, weather stations are subject to microclimate conditions and instrumentation varies between them. For cities in states such as Alaska, there are issues such as humidity that is affected by sources of moisture such as heavy soaking wood or standing water. In the visible figure showing the histograms, boxplot, and qq plot for percentage of possible sunshine and shows these figures after outliers are removed. There are noticeable enough outliers created by problems such as ice, debris, or even shadow that can affect solar radiation measurements which in turn affect the possibility of sunshine. The same method is applied to the other fourteen attributes. Removing outliers in our process does not change any critical values and is acceptable for our analysis.



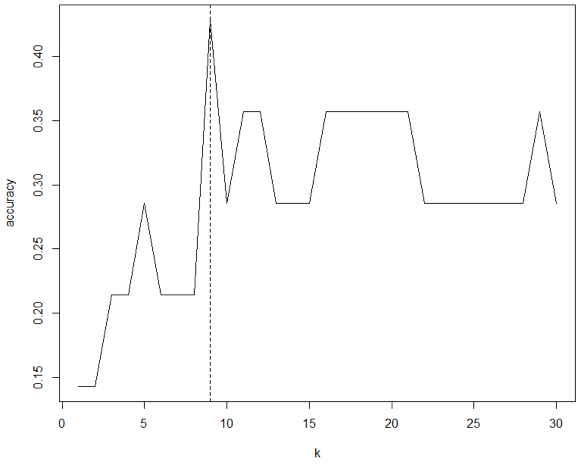
Results

Average Snowfall

In our analysis of average snowfall, there were many issues with finding an appropriate model. The assumptions of linear regression were violated, even after considering transformations. Even so, results from these models all showed that maximum humidity, mean days of precipitation greater than 0.01 inch, and lowest record temperature are the most significant factors related to average snowfall. Our generalized linear model using a gamma family distribution found significant covariates of highest record temperature, lowest record temperature, mean days of precipitation greater than 0.01 inch, and maximum humidity. After fitting the model with the training dataset, we tested its predictive accuracy on the test dataset, and achieved a RMSE = 1.459769. Our results from our logistic regression model with average snowfall as a binary variable were inconclusive. We were not able to find an adequately significant model, but our final model included average percent of possible sunshine, maximum wind speed, and record low temperature. Our RMSE achieved in predicting the test dataset was 9032.765, much larger than that of our model using a gamma distribution family and log link. Model diagnostics for our gamma and logistic generalized linear regression models also showed similar problems to our linear regression model. Because of the nature of weather data, it is also unlikely that we can assume independence of the observations, further impacting our conclusions. Due to the many violations of model assumptions, we conclude that linear regression and generalized linear models are inadequate to fully explore the relationship between snowfall and other climatic variables. Maximum humidity, mean days of precipitation greater than 0.01 inch, and lowest record temperature have the most significant relationship with average snowfall, and future models that focus on these variables in relation to snowfall should be considered. Spatial and temporal analysis of these variables is a future consideration that we did not cover in our analysis.

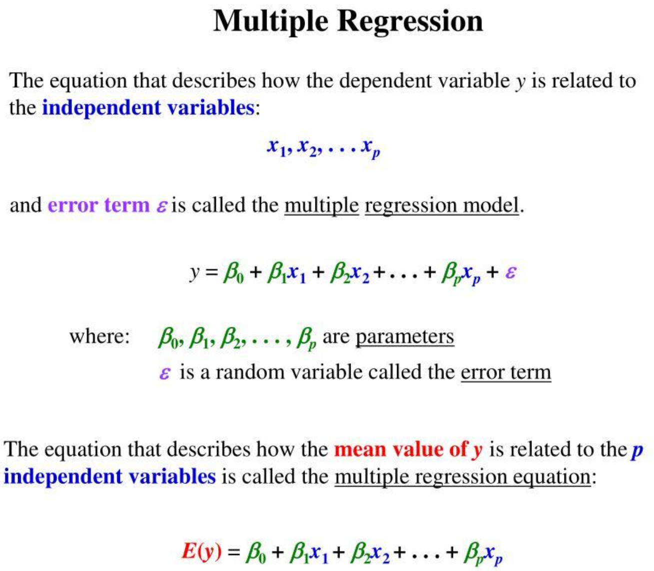
Regional Differences

In our analysis of regional differences, our kNN algorithm concluded that the 9 nearest observations produce the lowest misclassification error rates. The plot of accuracy vs. the different k values is provided. Still, the misclassification rate of the kNN model was just below 60%, which means there were many errors in classification. All weather stations in Hawaii were classified as the Southeast region of the United States. These stations had similar levels in percentage of possible sunshine and mean number of days with precipitation greater than 0.01 inch as those of the stations in the southeast. Stations in Hawaii still experience higher than the southeast region average temperature and lower than the southeast region average wind speed. Our kNN algorithm classified Alaska stations into both the northeast region and the midwest region of the United States. There were larger differences between the Alaska stations and those of the northeast and midwest: stations in Alaska experienced much lower temperatures and humidity, and higher average snowfall than both northeast and midwest stations.



Multilinear Regression Analysis

For this part of the project, a multilinear regression model were implemented to define one of the attributes as a linear function of the other 14 attributes. The response variable was chosen to be hghtmp.



Multiple Regression

The equation that describes how the dependent variable y is related to the **independent variables**:

$$x_1, x_2, \ldots x_p$$

and **error term** ε is called the multiple regression model.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$$

where: $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ are parameters

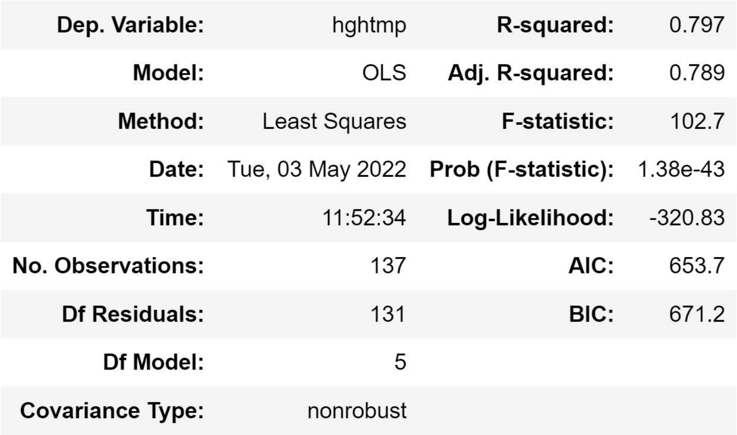
ε is a random variable called the error term

The equation that describes how the **mean value of y** is related to the p **independent variables** is called the multiple regression equation:

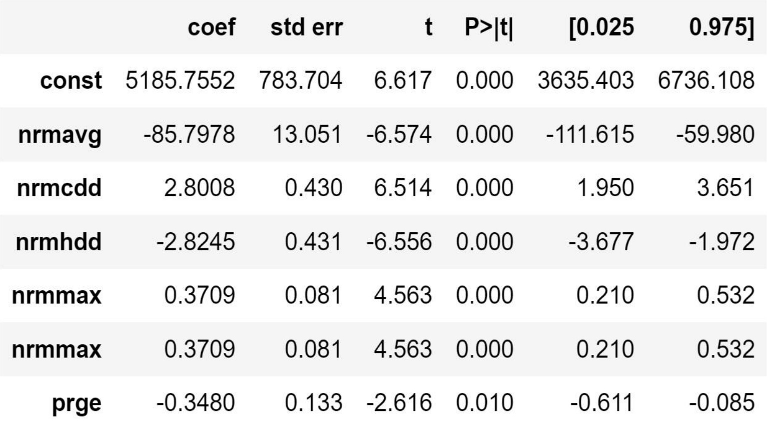
$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

After forming the model, the independed variables which were not contrinuting to the variability of the model and/or were a bad fit to the linear model were determined and dropped from the linear model. Checking the parameter estimates of the linear model, such as the p-values of the independent variables the R-squared, AIC and BIC scores, the best multilinear model were deteremined to be hghtmp variable versus the independed variables 'nrmavg','nrmcdd','nrmhdd','nrmmax','nrmmax','prge'.

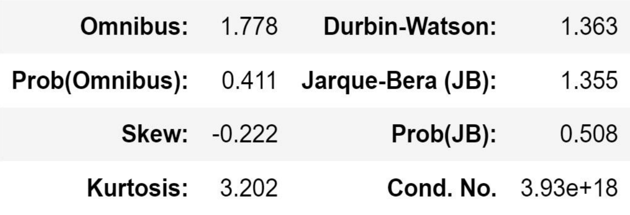
The summary results of the implentation of the model to the main dataset are brought in the following.



Dep. Variable:	hghtmp	**R-squared:**	0.797
Model:	OLS	**Adj. R-squared:**	0.789
Method:	Least Squares	**F-statistic:**	102.7
Date:	Tue, 03 May 2022	**Prob (F-statistic):**	1.38e-43
Time:	11:52:34	**Log-Likelihood:**	-320.83
No. Observations:	137	**AIC:**	653.7
Df Residuals:	131	**BIC:**	671.2
Df Model:	5		
Covariance Type:	nonrobust		



	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	5185.7552	783.704	6.617	0.000	3635.403	6736.108
nrmavg	-85.7978	13.051	-6.574	0.000	-111.615	-59.980
nrmcdd	2.8008	0.430	6.514	0.000	1.950	3.651
nrmhdd	-2.8245	0.431	-6.556	0.000	-3.677	-1.972
nrmmax	0.3709	0.081	4.563	0.000	0.210	0.532
nrmmax	0.3709	0.081	4.563	0.000	0.210	0.532
prge	-0.3480	0.133	-2.616	0.010	-0.611	-0.085



Omnibus:	1.778	**Durbin-Watson:**	1.363
Prob(Omnibus):	0.411	**Jarque-Bera (JB):**	1.355
Skew:	-0.222	**Prob(JB):**	0.508
Kurtosis:	3.202	**Cond. No.**	3.93e+18

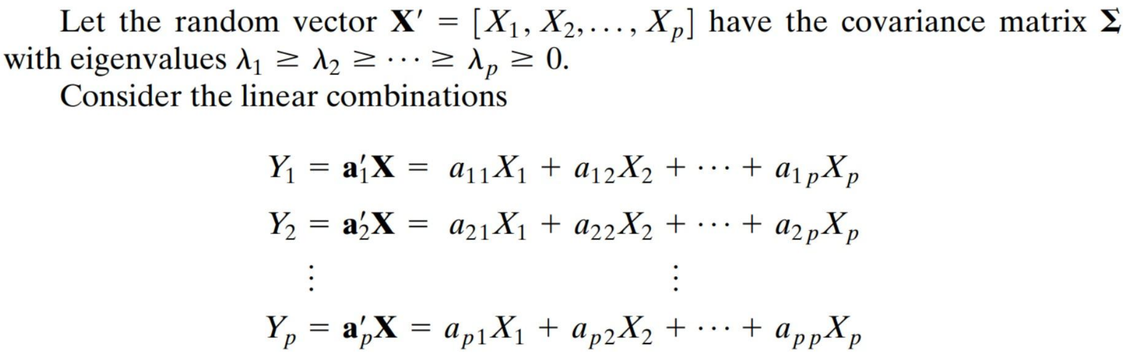
The multilinear model can now be defined in the form of the following linear equation:

Hghtmp = 5185.76 -85.80 nrmvg +2.80 nrmcdd -2.82nrmhdd +0.37nrmmax -0.35prge

The multilinear model can explain about 80 percent of the variabilty of the data in the dataset.

Principal Component Analysis

One of the most effective ways to reduce the dimentionality of the data is the Pricipal Component Analysis. In this method, a new set of variables are defined for the dataset such that all of these variables are othogonal to each other and as the direct result, there will be no colinearity between each two of them. The following decreption depicts the recipe of how the Pricipal Comonents are defined and how the variances of the data (the eigen values) are computed.



Let the random vector $\mathbf{X}' = [X_1, X_2, \ldots, X_p]$ have the covariance matrix $\mathbf{\Sigma}$ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$.

Consider the linear combinations

$$\begin{aligned} Y_1 &= \mathbf{a}_1'\mathbf{X} = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p \\ Y_2 &= \mathbf{a}_2'\mathbf{X} = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p \\ &\vdots \qquad\qquad\qquad\qquad \vdots \\ Y_p &= \mathbf{a}_p'\mathbf{X} = a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p \end{aligned}$$

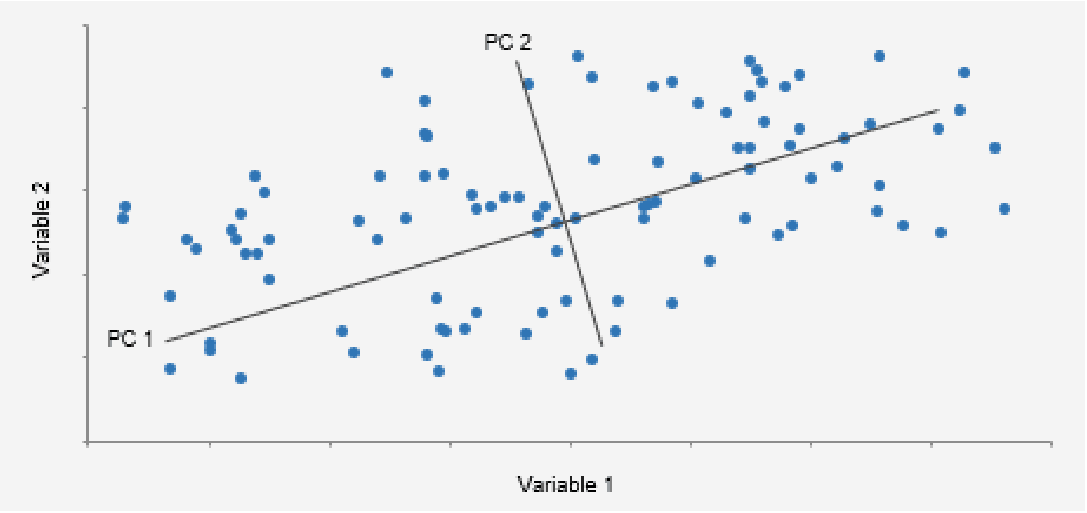
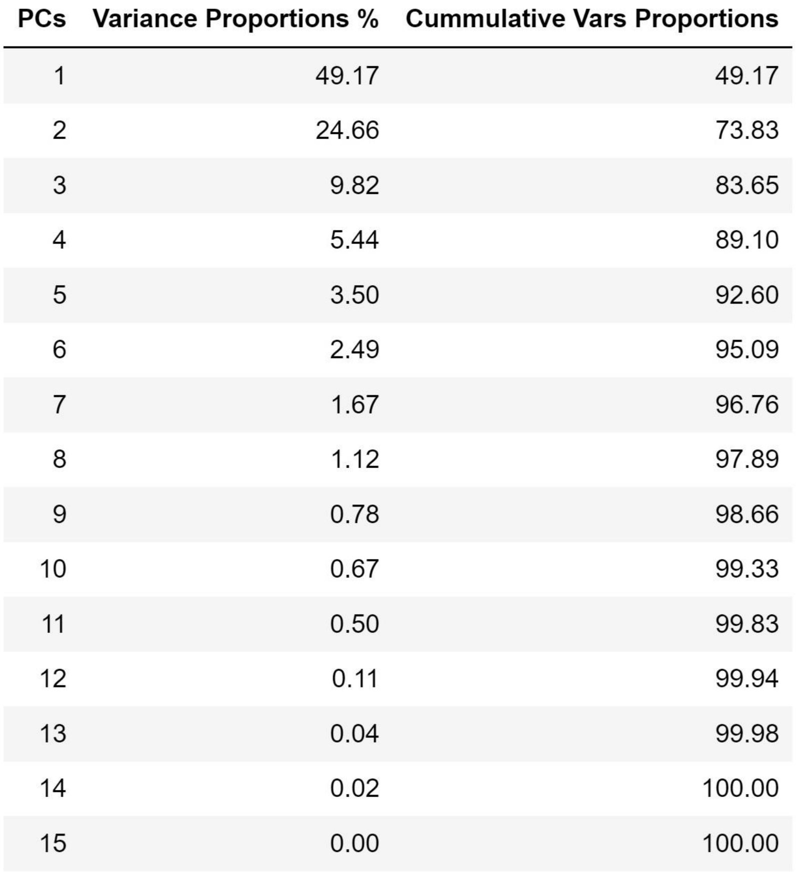


Figure 4. The pricipal components are oriented along the largest spread of the data

Also, in this method the principal components are oriented along the largest spread (variabilty) of the datapoints. Subject to the need of the analysis, only the first few of the components are chosen and the dataset values are computed to represent the coordinate of the data points along these newly defined “Pricipal” components.



PCs	Variance Proportions %	Cummulative Vars Proportions
1	49.17	49.17
2	24.66	73.83
3	9.82	83.65
4	5.44	89.10
5	3.50	92.60
6	2.49	95.09
7	1.67	96.76
8	1.12	97.89
9	0.78	98.66
10	0.67	99.33
11	0.50	99.83
12	0.11	99.94
13	0.04	99.98
14	0.02	100.00
15	0.00	100.00

Figure 5. The table of the proportion of the data explained by each Principal Component.

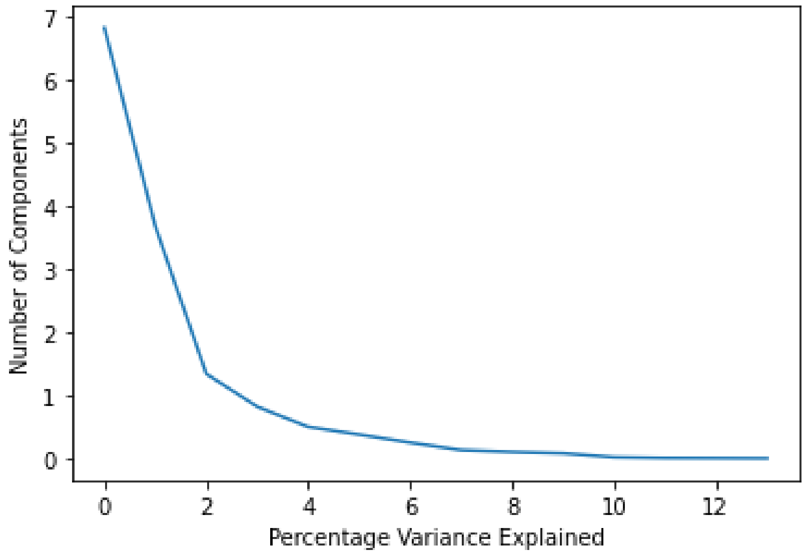
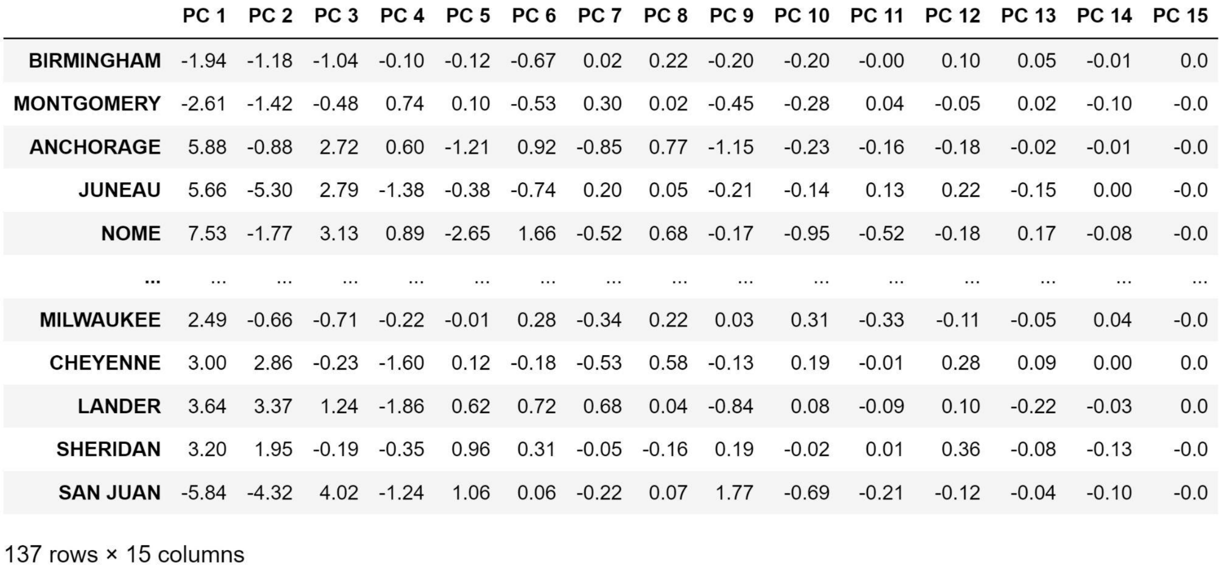


Figure 6. The more Principal Components are defined, the less the variance of the data is explained along the added PCs.

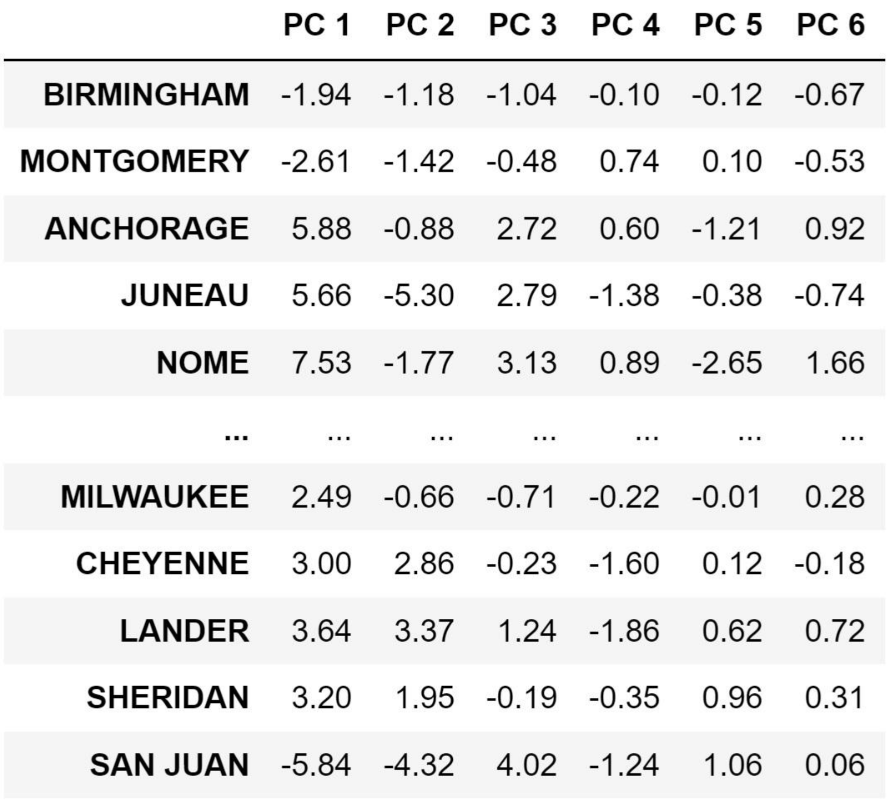


	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6	PC 7	PC 8	PC 9	PC 10	PC 11	PC 12	PC 13	PC 14	PC 15
BIRMINGHAM	-1.94	-1.18	-1.04	-0.10	-0.12	-0.67	0.02	0.22	-0.20	-0.20	-0.00	0.10	0.05	-0.01	0.0
MONTGOMERY	-2.61	-1.42	-0.48	0.74	0.10	-0.53	0.30	0.02	-0.45	-0.28	0.04	-0.05	0.02	-0.10	-0.0
ANCHORAGE	5.88	-0.88	2.72	0.60	-1.21	0.92	-0.85	0.77	-1.15	-0.23	-0.16	-0.18	-0.02	-0.01	-0.0
JUNEAU	5.66	-5.30	2.79	-1.38	-0.38	-0.74	0.20	0.05	-0.21	-0.14	0.13	0.22	-0.15	0.00	-0.0
NOME	7.53	-1.77	3.13	0.89	-2.65	1.66	-0.52	0.68	-0.17	-0.95	-0.52	-0.18	0.17	-0.08	-0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
MILWAUKEE	2.49	-0.66	-0.71	-0.22	-0.01	0.28	-0.34	0.22	0.03	0.31	-0.33	-0.11	-0.05	0.04	-0.0
CHEYENNE	3.00	2.86	-0.23	-1.60	0.12	-0.18	-0.53	0.58	-0.13	0.19	-0.01	0.28	0.09	0.00	0.0
LANDER	3.64	3.37	1.24	-1.86	0.62	0.72	0.68	0.04	-0.84	0.08	-0.09	0.10	-0.22	-0.03	0.0
SHERIDAN	3.20	1.95	-0.19	-0.35	0.96	0.31	-0.05	-0.16	0.19	-0.02	0.01	0.36	-0.08	-0.13	-0.0
SAN JUAN	-5.84	-4.32	4.02	-1.24	1.06	0.06	-0.22	0.07	1.77	-0.69	-0.21	-0.12	-0.04	-0.10	-0.0

137 rows × 15 columns

Figure 7. The whole data set is defined along the Principal Components (PC). The coordinates of the data point are computed along each PC.

It turned out that the first 6 PCs can explain around 95% of the information (variability) stored in the data. That is around one third of the original dimentionality of the data. The projection of the data along the first 6 PCs are brought in the following data table.



	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6
BIRMINGHAM	-1.94	-1.18	-1.04	-0.10	-0.12	-0.67
MONTGOMERY	-2.61	-1.42	-0.48	0.74	0.10	-0.53
ANCHORAGE	5.88	-0.88	2.72	0.60	-1.21	0.92
JUNEAU	5.66	-5.30	2.79	-1.38	-0.38	-0.74
NOME	7.53	-1.77	3.13	0.89	-2.65	1.66
...	...	...	...	...	...	...
MILWAUKEE	2.49	-0.66	-0.71	-0.22	-0.01	0.28
CHEYENNE	3.00	2.86	-0.23	-1.60	0.12	-0.18
LANDER	3.64	3.37	1.24	-1.86	0.62	0.72
SHERIDAN	3.20	1.95	-0.19	-0.35	0.96	0.31
SAN JUAN	-5.84	-4.32	4.02	-1.24	1.06	0.06

Figure 8. The coordinates of the data point along the first 6 Principal Components.

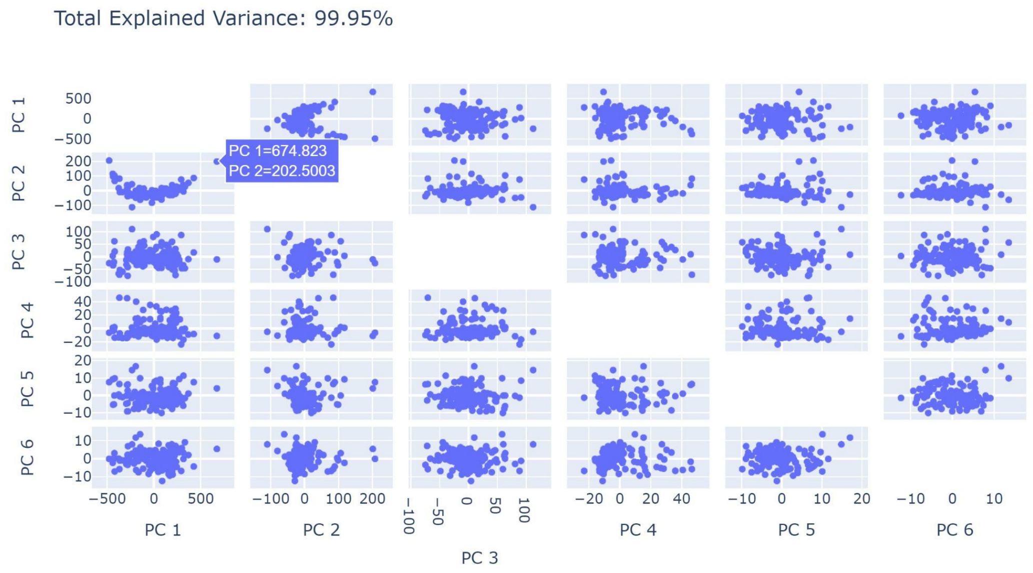


Figure 9. The data points are projected on a two-by-two PCs

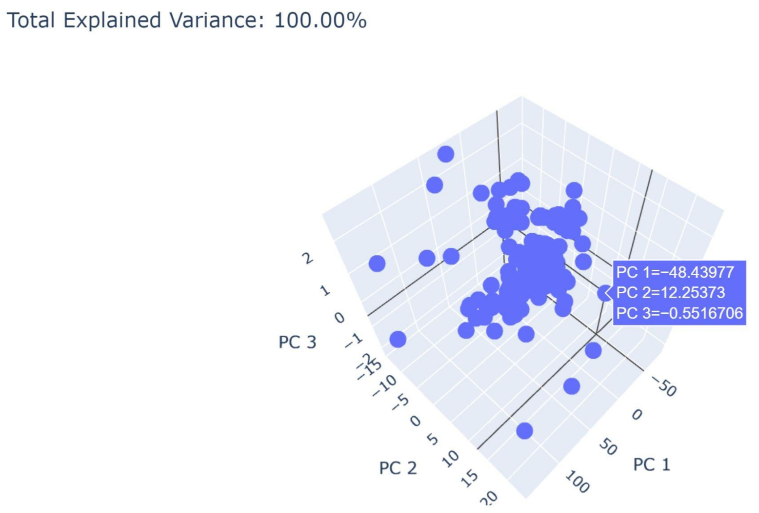
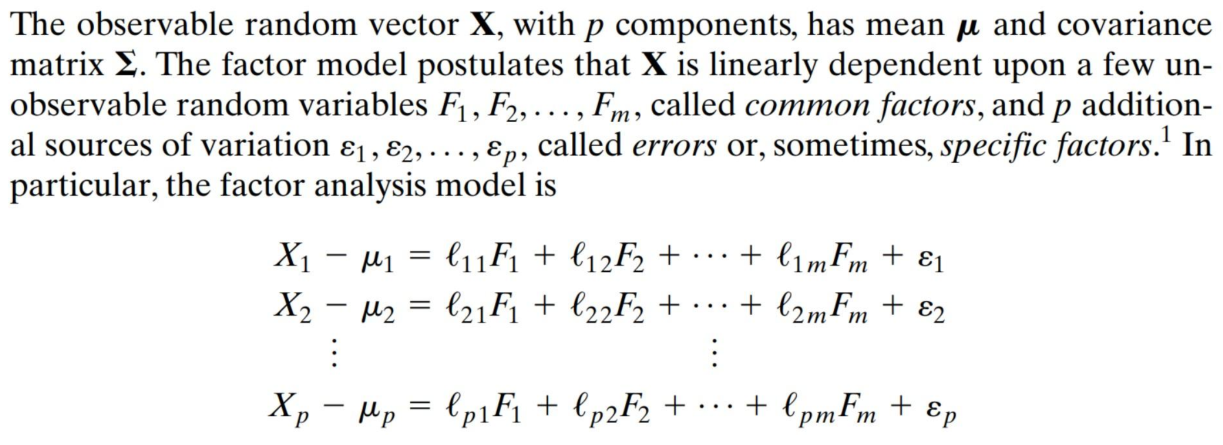


Figure 10. The orientation of the complete set of data points in 3-dimentional space defined by the first 3 PCs.

Factor Analysis

In the statistical analysis, sometimes, there are hidden or lattent variables can be considered that are not measured or (sometimes) can’t be measured quantitavely.These hidden variables can be defined as the linear combination of the presented variables in the data set. In fact, The factor analysis technique extracts the maximum common variance from all the variables and puts them into a common score. It is a theory that is used in training the machine learning model and so it is quite related to data mining. The following decribes how the these lattent variables or the Factors are defined:



The observable random vector $\mathbf{X}$, with p components, has mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The factor model postulates that $\mathbf{X}$ is linearly dependent upon a few unobservable random variables $F_1, F_2, \ldots, F_m$, called *common factors*, and p additional sources of variation $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p$, called *errors* or, sometimes, *specific factors*.[1] In particular, the factor analysis model is

$$\begin{aligned}
X_1 - \mu_1 &= \ell_{11}F_1 + \ell_{12}F_2 + \cdots + \ell_{1m}F_m + \varepsilon_1 \\
X_2 - \mu_2 &= \ell_{21}F_1 + \ell_{22}F_2 + \cdots + \ell_{2m}F_m + \varepsilon_2 \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
X_p - \mu_p &= \ell_{p1}F_1 + \ell_{p2}F_2 + \cdots + \ell_{pm}F_m + \varepsilon_p
\end{aligned}$$

In order to apply Factor Analysis techniques, one can appy the Bartlett's test. Bartlett's test of sphericity checks whether or not the observed variables intercorrelate at all using the observed correlation matrix against the identity matrix. If the test found statistically insignificant, you should not employ a factor analysis. Kaiser-Meyer-Olkin (KMO) Test measures the suitability of data for factor analysis. It determines the adequacy for each observed variable and for the complete model. KMO estimates the proportion of variance among all the observed variable. Lower proportion id more suitable for factor analysis. KMO values range between 0 and 1. Value of KMO less than 0.6 is considered inadequate. In our analysis the KMO value is 0.7495.

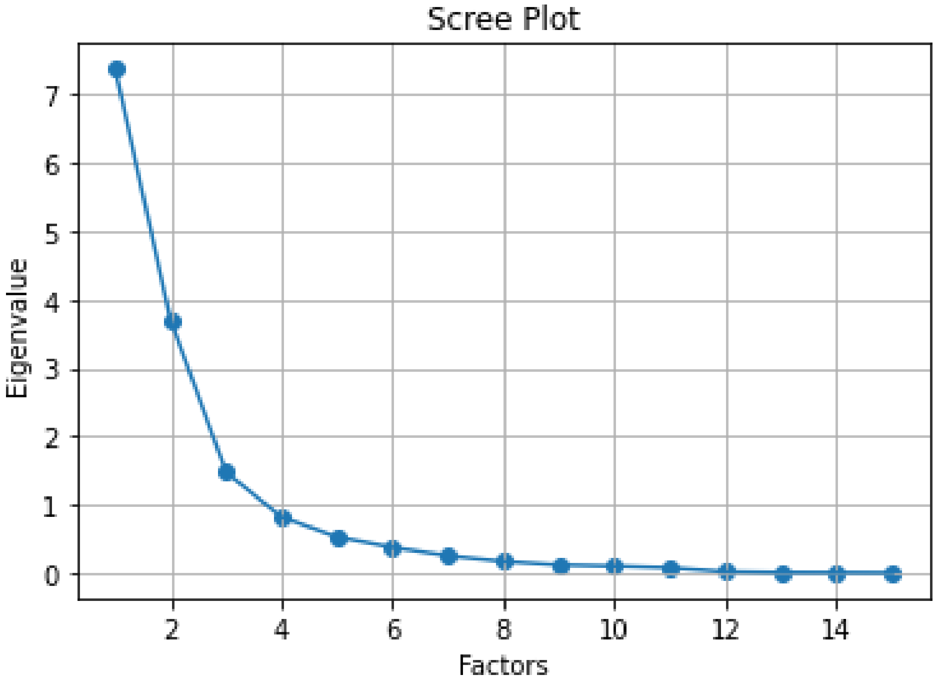
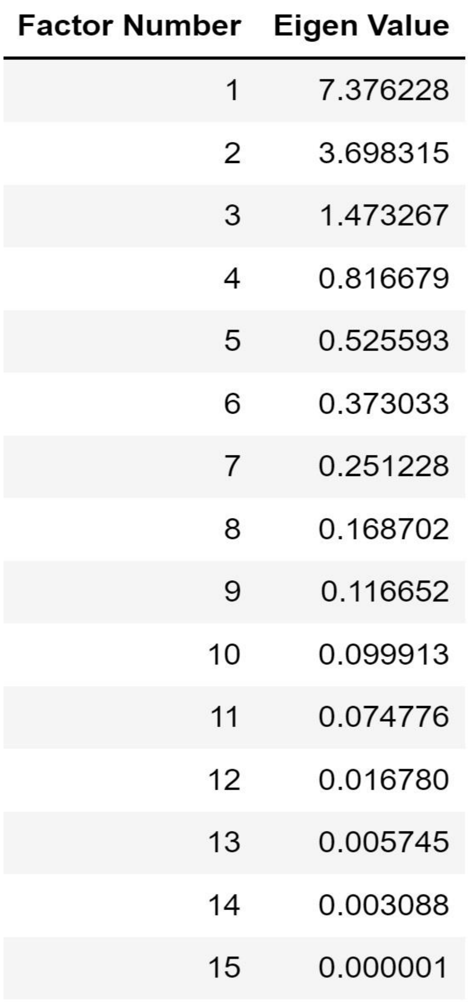


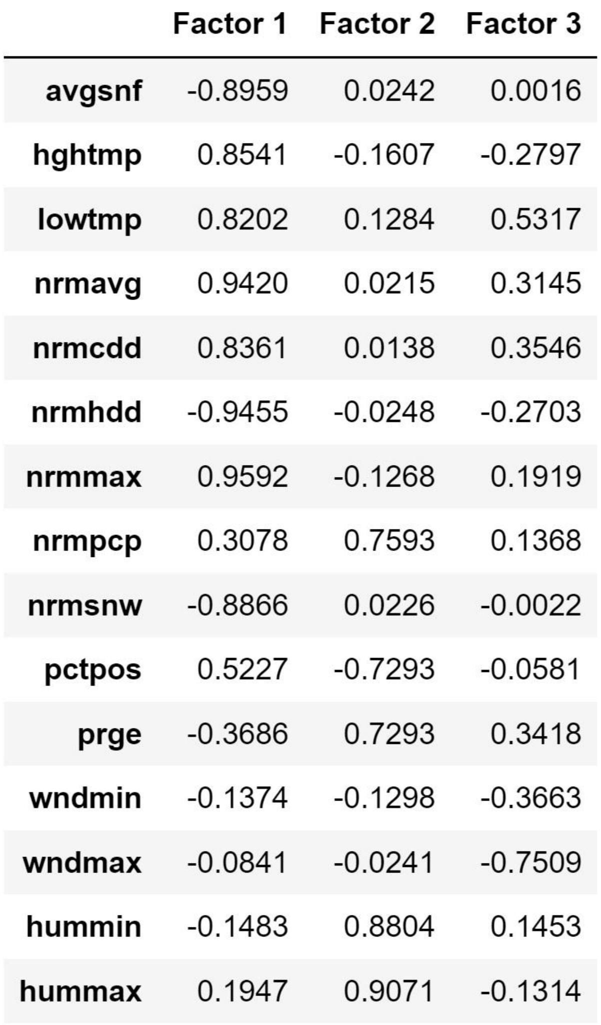
Figure 11. In this graph, the eigenvalues associated to each computed factors are presented.



Factor Number	Eigen Value
1	7.376228
2	3.698315
3	1.473267
4	0.816679
5	0.525593
6	0.373033
7	0.251228
8	0.168702
9	0.116652
10	0.099913
11	0.074776
12	0.016780
13	0.005745
14	0.003088
15	0.000001

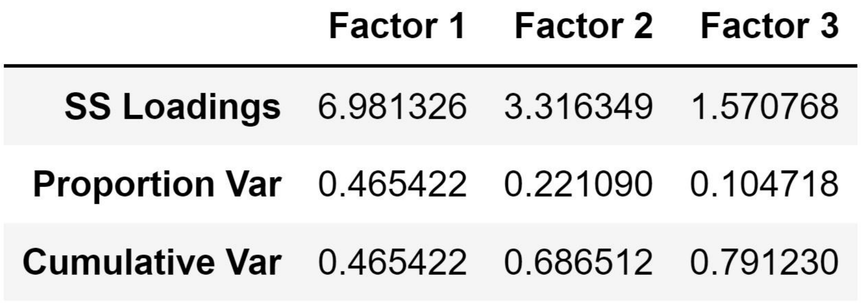
Figure 12. The computed factors and their associated eigenvalues

For the Factor Analysis, only those factors with eigenvalues of larger than 1 are selected. In our analysis only three factors are selected. After that, score or the coordinates of each of the original variables (the 15 climate attributes) are computed against the first three factors. Each factor with the highest score for each original variables are the most effective ones in those particular variables. The following two tables show the summary of the factor analysis for the first three factors computed from the main dataset.



	Factor 1	Factor 2	Factor 3
avgsnf	-0.8959	0.0242	0.0016
hghtmp	0.8541	-0.1607	-0.2797
lowtmp	0.8202	0.1284	0.5317
nrmavg	0.9420	0.0215	0.3145
nrmcdd	0.8361	0.0138	0.3546
nrmhdd	-0.9455	-0.0248	-0.2703
nrmmax	0.9592	-0.1268	0.1919
nrmpcp	0.3078	0.7593	0.1368
nrmsnw	-0.8866	0.0226	-0.0022
pctpos	0.5227	-0.7293	-0.0581
prge	-0.3686	0.7293	0.3418
wndmin	-0.1374	-0.1298	-0.3663
wndmax	-0.0841	-0.0241	-0.7509
hummin	-0.1483	0.8804	0.1453
hummax	0.1947	0.9071	-0.1314

Figure 14. The scores of the 15 attributes for the three Factors.



	Factor 1	Factor 2	Factor 3
SS Loadings	6.981326	3.316349	1.570768
Proportion Var	0.465422	0.221090	0.104718
Cumulative Var	0.465422	0.686512	0.791230

Figure 15. The SS loading scores, proportion of the variances explained and cummulative variances explained by each foctor for the main data set.

Conclusions & Limitations

The most glaring limitation of these methods and results stem from the averaging process over all the data. This process was the product of good intent; it was a consensus-driven decision at the onset of this report that this technique would simplify these analyses, account for missing data, and serve as a viable method to organize the data in an easier-to-use format. Although it made workflow considerably easier, a lot of statistically-pertinent information was lost over the course of the process. Further, the NOAA strictly states on its website these data should not be used for in-depth analysis, but rather as a reference tool.

The United States can be divided into any given number of regions depending on the question(s) being asked. For the purposes of this analysis, the regions were formulated based on what were felt to be the most important climatic division(s). For example, the justification for PC was its proximity to coastal waters. However, given the large variability of climate measures within PC, this decision proved to be a bit inappropriate. In the future, algorithmic clustering techniques can be implemented to statistically "place" US cities into statistically significant regions based on their measures; from this, further and more meaningful results can be inferred. For example, socioeconomic status (SES) measures can be implemented in order to determine if there is in fact a relationship between those areas with higher SES and "nicer" weather, i.e. to determine whether the so-called "Sunshine Tax" ideology holds true.

Our constructed models for average snowfall experienced many issues with model assumptions. Different methods for analyzing spatial and temporal patterns in this dataset were not fully explored due to unfamiliarity. A future analysis of climatological data would require this exploration.

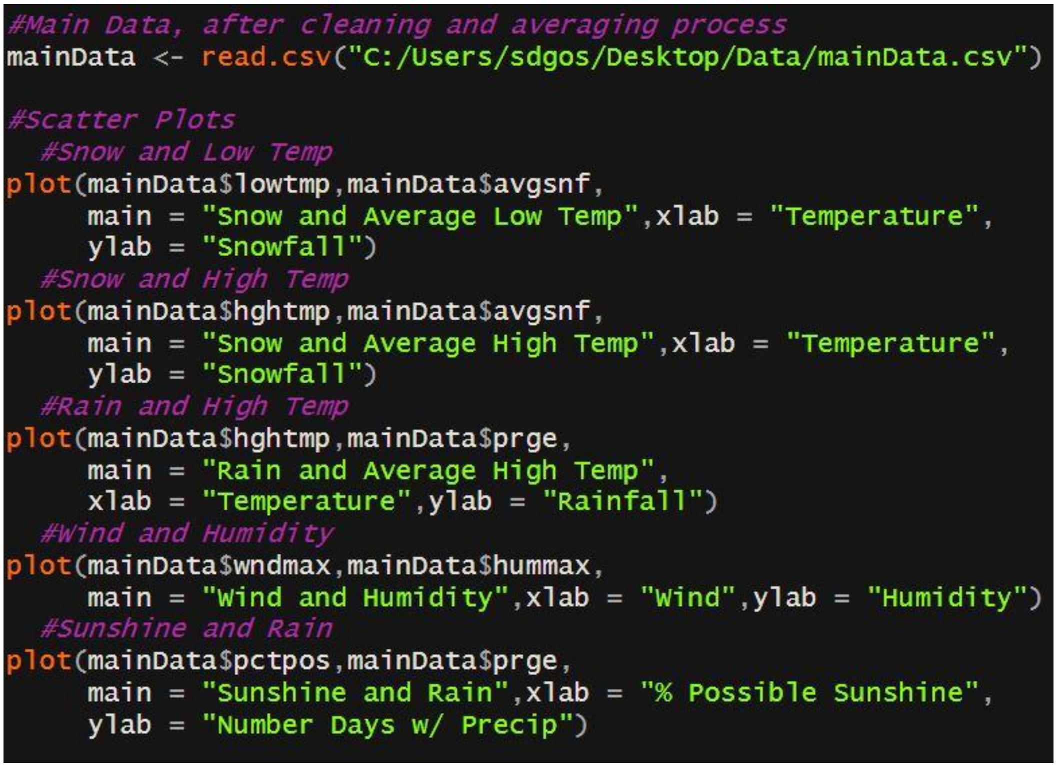
References

- Index of /pub/Data/CCD-data. (n.d.). Retrieved May 17, 2022, from https://www1.ncdc.noaa.gov/pub/data/ccd-data/

Appendix

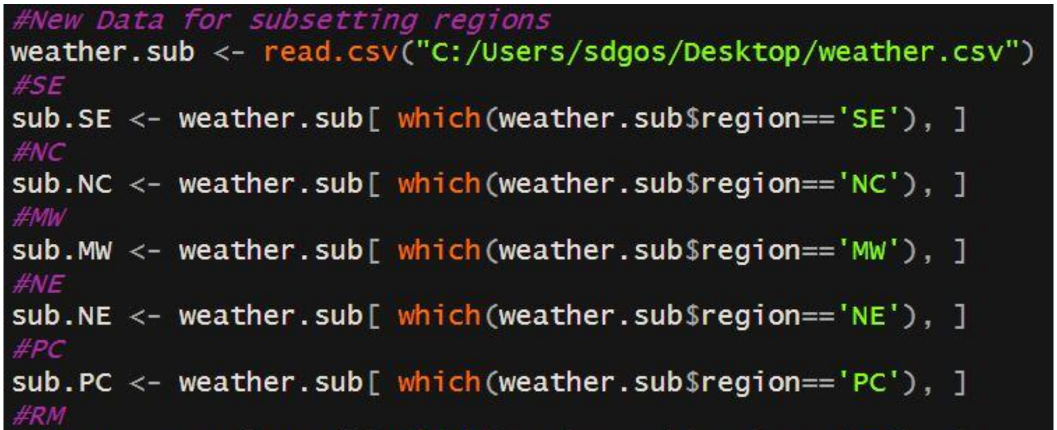
EDA: The following are R-code samples for the EDA process. This is not an exhaustive breakdown of each code block, but rather a sample of general workflow. Oftentimes, variables would be changed to produce output with the same block of code, hence its straightforwardness.

1- **Scatter-plot code to show relationships between different covariates.**



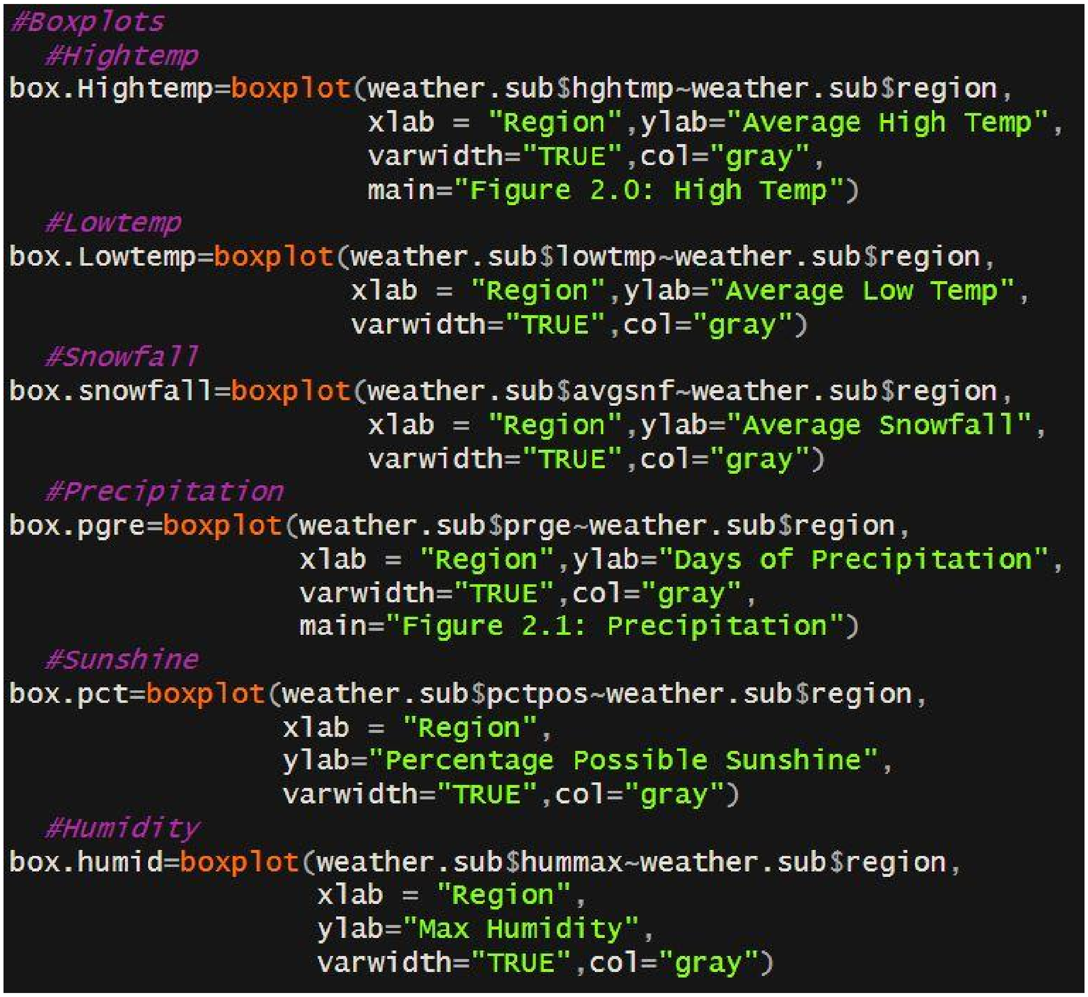
```r
#Main Data, after cleaning and averaging process
mainData <- read.csv("C:/Users/sdgos/Desktop/Data/mainData.csv")

#Scatter Plots
  #Snow and Low Temp
plot(mainData$lowtmp,mainData$avgsnf,
     main = "Snow and Average Low Temp",xlab = "Temperature",
     ylab = "Snowfall")
  #Snow and High Temp
plot(mainData$hghtmp,mainData$avgsnf,
     main = "Snow and Average High Temp",xlab = "Temperature",
     ylab = "Snowfall")
  #Rain and High Temp
plot(mainData$hghtmp,mainData$prge,
     main = "Rain and Average High Temp",
     xlab = "Temperature",ylab = "Rainfall")
  #Wind and Humidity
plot(mainData$wndmax,mainData$hummax,
     main = "Wind and Humidity",xlab = "Wind",ylab = "Humidity")
  #Sunshine and Rain
plot(mainData$pctpos,mainData$prge,
     main = "Sunshine and Rain",xlab = "% Possible Sunshine",
     ylab = "Number Days w/ Precip")
```

2- This code block shows the grouping into the various “regions”



```r
#New Data for subsetting regions
weather.sub <- read.csv("C:/Users/sdgos/Desktop/weather.csv")
#SE
sub.SE <- weather.sub[ which(weather.sub$region=='SE'), ]
#NC
sub.NC <- weather.sub[ which(weather.sub$region=='NC'), ]
#MW
sub.MW <- weather.sub[ which(weather.sub$region=='MW'), ]
#NE
sub.NE <- weather.sub[ which(weather.sub$region=='NE'), ]
#PC
sub.PC <- weather.sub[ which(weather.sub$region=='PC'), ]
#RM
```

3- Once regions had been created, boxplots could then be created. This code block illustrates that general workflow.

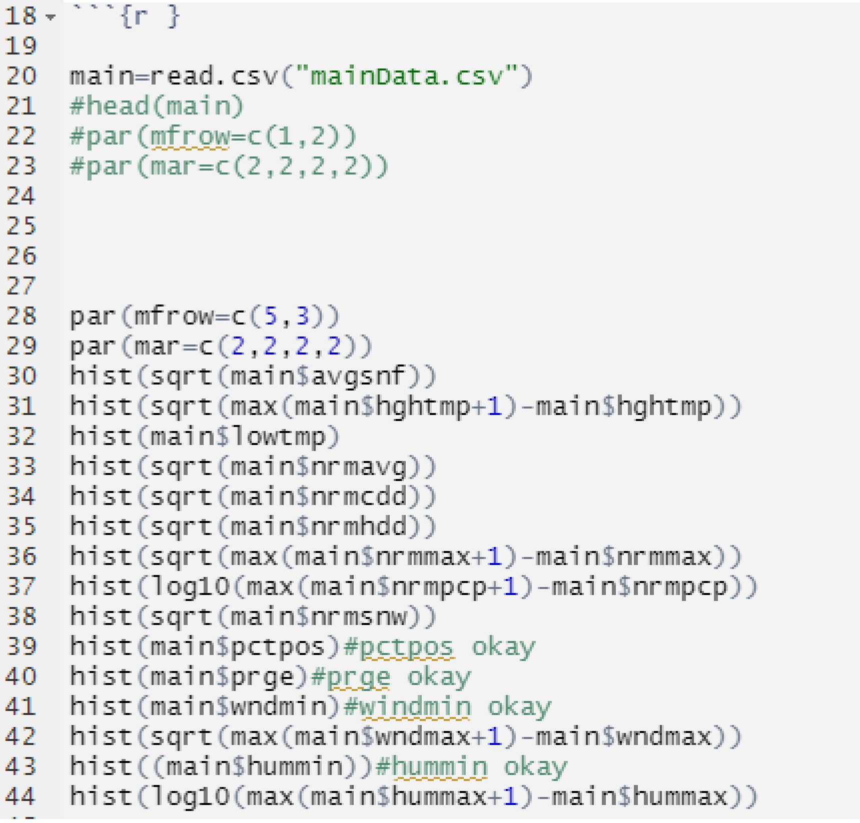


```
#Boxplots
  #Hightemp
box.Hightemp=boxplot(weather.sub$hghtmp~weather.sub$region,
                     xlab = "Region",ylab="Average High Temp",
                     varwidth="TRUE",col="gray",
                     main="Figure 2.0: High Temp")
  #Lowtemp
box.Lowtemp=boxplot(weather.sub$lowtmp~weather.sub$region,
                    xlab = "Region",ylab="Average Low Temp",
                    varwidth="TRUE",col="gray")
  #Snowfall
box.snowfall=boxplot(weather.sub$avgsnf~weather.sub$region,
                     xlab = "Region",ylab="Average Snowfall",
                     varwidth="TRUE",col="gray")
  #Precipitation
box.pgre=boxplot(weather.sub$prge~weather.sub$region,
                 xlab = "Region",ylab="Days of Precipitation",
                 varwidth="TRUE",col="gray",
                 main="Figure 2.1: Precipitation")
  #Sunshine
box.pct=boxplot(weather.sub$pctpos~weather.sub$region,
                xlab = "Region",
                ylab="Percentage Possible Sunshine",
                varwidth="TRUE",col="gray")
  #Humidity
box.humid=boxplot(weather.sub$hummax~weather.sub$region,
                  xlab = "Region",
                  ylab="Max Humidity",
                  varwidth="TRUE",col="gray")
```

Normalization:

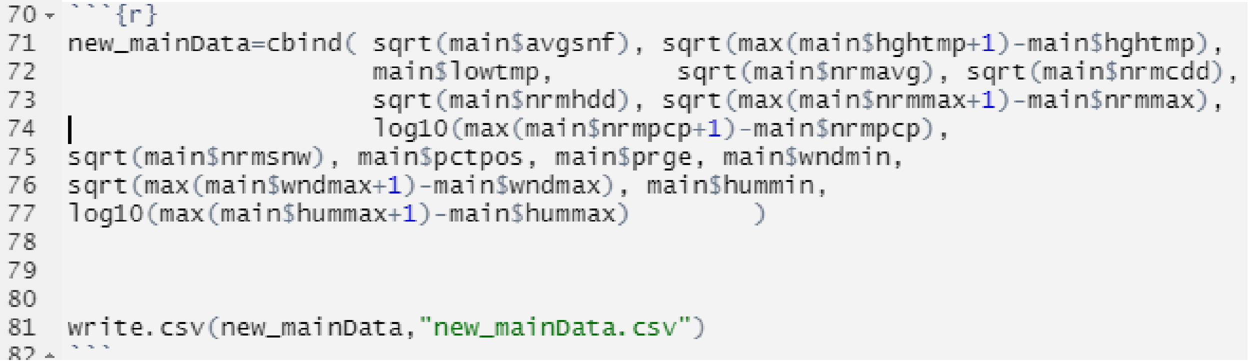
Following code comes from mainData.csv run through R markdown. The code providing for this section shows how histogram figures are established through data normalization and how we create figures following the removal of outliers.

Data comes from mainData.csv where we end up establishing 15 histogram figures for each of the attributes. These are the histograms we have after transforming them toward the central tendency for normalization using primarily square root and natural log transformation.



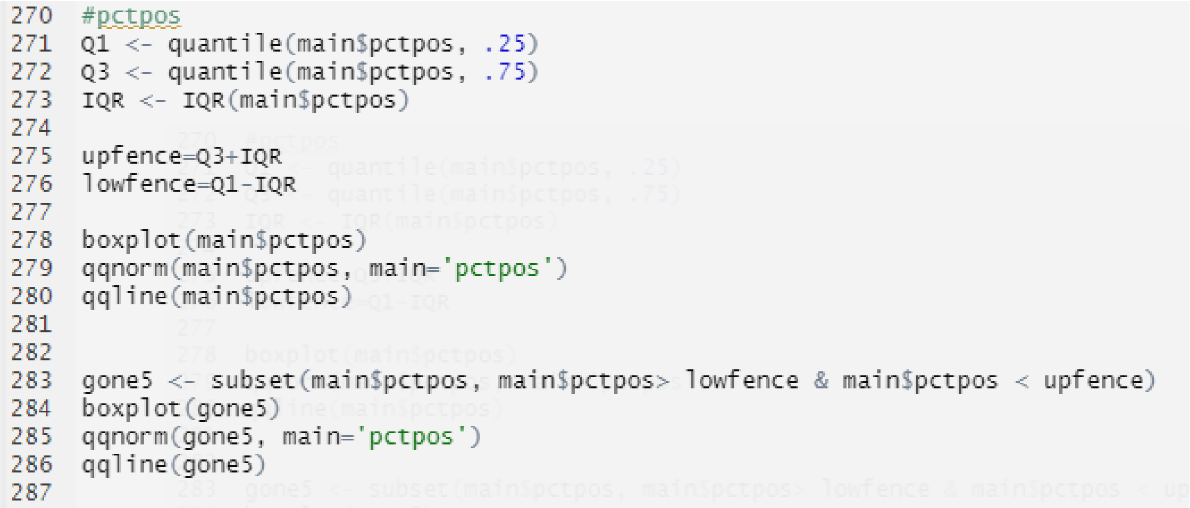
~~~
```{r }

main=read.csv("mainData.csv")
#head(main)
#par(mfrow=c(1,2))
#par(mar=c(2,2,2,2))




par(mfrow=c(5,3))
par(mar=c(2,2,2,2))
hist(sqrt(main$avgsnf))
hist(sqrt(max(main$hghtmp+1)-main$hghtmp))
hist(main$lowtmp)
hist(sqrt(main$nrmavg))
hist(sqrt(main$nrmcdd))
hist(sqrt(main$nrmhdd))
hist(sqrt(max(main$nrmmax+1)-main$nrmmax))
hist(log10(max(main$nrmpcp+1)-main$nrmpcp))
hist(sqrt(main$nrmsnw))
hist(main$pctpos)#pctpos okay
hist(main$prge)#prge okay
hist(main$wndmin)#windmin okay
hist(sqrt(max(main$wndmax+1)-main$wndmax))
hist((main$hummin))#hummin okay
hist(log10(max(main$hummax+1)-main$hummax))
~~~

Normalized data is now stored in a new CSV file

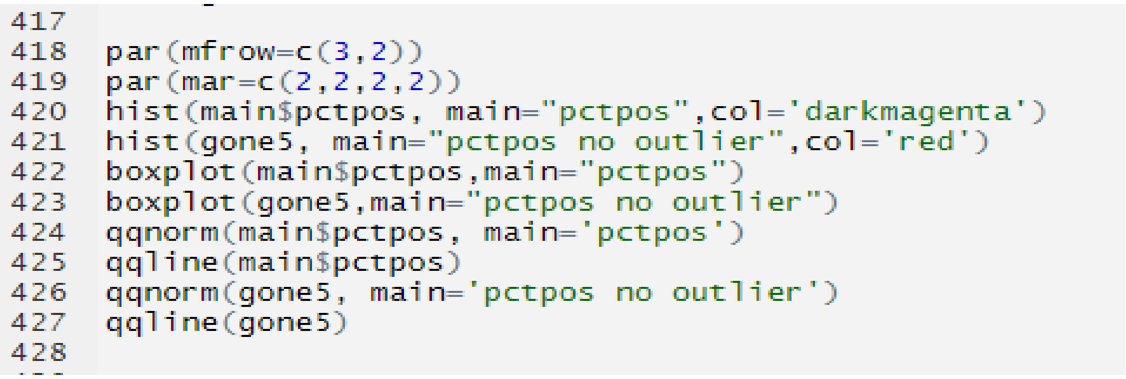


~~~
```{r}
new_mainData=cbind( sqrt(main$avgsnf), sqrt(max(main$hghtmp+1)-main$hghtmp),
                    main$lowtmp,        sqrt(main$nrmavg), sqrt(main$nrmcdd),
                    sqrt(main$nrmhdd), sqrt(max(main$nrmmax+1)-main$nrmmax),
                    log10(max(main$nrmpcp+1)-main$nrmpcp),
sqrt(main$nrmsnw), main$pctpos, main$prge, main$wndmin,
sqrt(max(main$wndmax+1)-main$wndmax), main$hummin,
log10(max(main$hummax+1)-main$hummax)        )



write.csv(new_mainData,"new_mainData.csv")
```
~~~

This code below provides the figures for normalized histogram, boxplot, and qqplot for percentage of possible sunshine both before and after outliers are removed. The code provided has the process for removing outliers. The same code is applied toward the other 14 attributes as well (substitute the variable name as the loop had problems).



```
#pctpos
Q1 <- quantile(main$pctpos, .25)
Q3 <- quantile(main$pctpos, .75)
IQR <- IQR(main$pctpos)

upfence=Q3+IQR
lowfence=Q1-IQR

boxplot(main$pctpos)
qqnorm(main$pctpos, main='pctpos')
qqline(main$pctpos)


gone5 <- subset(main$pctpos, main$pctpos> lowfence & main$pctpos < upfence)
boxplot(gone5)
qqnorm(gone5, main='pctpos')
qqline(gone5)

```



```

par(mfrow=c(3,2))
par(mar=c(2,2,2,2))
hist(main$pctpos, main="pctpos",col='darkmagenta')
hist(gone5, main="pctpos no outlier",col='red')
boxplot(main$pctpos,main="pctpos")
boxplot(gone5,main="pctpos no outlier")
qqnorm(main$pctpos, main='pctpos')
qqline(main$pctpos)
qqnorm(gone5, main='pctpos no outlier')
qqline(gone5)

```

R code: Snowfall and Regional Differences

library(MASS)
library(ggplot2)

```
library(dplyr)
library(caret)
library(class)
library(car)

# Read data
data = read.csv("weather.csv",header = T)
# Remove normal constant variables and non-numerical variables
data = data[,-c(1:4,8:13)]
# center and scale
for (i in 1:ncol(data)) {
  data[,i] = scale(data[,i],center = T,scale = T)
}

## Average Snowfall
### Linear Regression

fit.reg = lm(avgsnf~.,data = data) # full non-transformed model
summary(fit.reg) # model
plot(fit.reg) # diagnostics

BC = boxcox(lm((avgsnf+1)~.,data = data)) # Box-Cox transformation
lambda = BC$x[which.max(BC$y)] # Retrieve best lambda
fit.reg = lm(((((avgsnf+1)^lambda-1)/lambda))~., data = data) # full model with Box-Cox transformation
fit.reg = lm(((((avgsnf+1)^lambda-1)/lambda))~hghtmp+lowtmp+pctpos+prge+wndmin+wndmax+hummax, data = data)
vif(fit.reg) # Remove hummin due to large VIF
step(fit.reg) # Stepwise

fit.reg = lm(((((avgsnf+1)^lambda-1)/lambda))~lowtmp+prge+hummax, data = weatherN) # Final Model Transformed Regression
summary(fit.reg)
plot(fit.reg) # diagnostics

### GLM

data = data.frame(data, snwfll = as.factor(ifelse(data$avgsnf < .02,0,1))) # Create snowfall variable
train = sample(seq_len(nrow(snowfall)),109) # training and test set, 10%
training = snowfall[train,]
test = snowfall[-train,]

#### Gamma family log link

fit.gamma = glm((avgsnf+1) ~ hghtmp+lowtmp+pctpos+prge+wndmax+wndmin+hummax+hummin,data = training, family = Gamma(link = "log")) # Full model
vif(fit.gamma)
step(fit.gamma)

fit.gamma = glm((avgsnf+1) ~ hghtmp+lowtmp+prge+hummax,data = training, family = Gamma(link = "log"))

summary(fit.gamma) # model
plot(fit.gamma) # diagnostics

predictions = predict(fit.gamma, newdata=test) # Test set predictions
testRMSE = sqrt(mean((predictions-test$avgsnf)^2)) #RMSE


##### Binomial logit link

fit.bin = glm(snwfll ~ hghtmp+lowtmp+pctpos+prge+wndmax+wndmin+hummax+hummin,data = training, family = binomial) # Full model
vif(fit.bin) # Major multicollinearity issues
step(fit.bin)

fit.bin = glm(snwfll ~ pctpos+wndmax+lowtmp,data = training, family = binomial)
summary(fit.bin)
plot(fit.bin)

predictions = predict(fit.bin, newdata=test) # Test set predictions
testRMSE = sqrt(mean((predictions-test$avgsnf)^2)) #RMSE


## Regional Differences
data = read.csv("weather.csv",header = T)
weather = data

min_max_norm <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

data = data[,-c(8,9,10,11,12,13)]
data$temp = (data$hghtmp-data$lowtmp)/2
data$hum = (data$hummax-data$hummin)/2
```

```
data$wind = (data$wndmax-data$wndmin)/2
data = data[,-c(10:13)]
data = as.data.frame(cbind(data[,2:4],lapply(data[5:11], min_max_norm))) # Normalize

fullKNN = data[data$region!="NC",]
full.samp = sample(seq_len(nrow(fullKNN)),14)
testfull = fullKNN[full.samp,]
trainfull = fullKNN[-full.samp,]

r = data.frame(array(NA, dim = c(0, 2), dimnames = list(NULL, c("k","accuracy"))))
for (k in 1:30) {

  # fit model on training set, predict test set data
  set.seed(60402)
  predictions <- knn(train = trainfull[,-c(1:3)],test = testfull[,-c(1:3)],cl = trainfull[,3],k=k)

  # confusion matrix on test set
  t = table(pred = predictions, ref = testfull[,3])

  # accuracy
  a = sum(diag(t)) / sum(t)

  # bind
  r = rbind(r, data.frame(k = k, accuracy = a))
}

r
r[which.max(r$accuracy),]

(k.best = r[which.max(r$accuracy),"k"])

with(r, plot(k, accuracy, type = "l")) # plot
abline(v = k.best, lty = 2)

full.knn = knn(train = trainfull[,-c(1:3)],test = testfull[,-c(1:3)],cl = trainfull[,3],k=9)
summary(full.knn)


#### Classify Hawaii and Alaska

train = data[-c(3,4,5,26,27,28),] # training and test set
test = normKNN[c(3,4,5,26,27,28),]

knn.nc = knn(train = train[,-c(1:3)],test = test[,-c(1:3)],cl = train[,3],k=9)
summary(knn.nc)
```

The codes are writted in Python for the pupose of data cleaning, preprocessing the the cleaned data, creating new data sets from the them followed by Linear Regression Analysis, Pricipal Componenet Analysis and Factor Analysis.

```
import os
import csv

import numpy          as np
import pandas         as pd
import matplotlib
import matplotlib.pyplot as plt
import pylab          as pyl
import plotly.express   as px
import statsmodels.api   as sm

from   sklearn                 import preprocessing
from   sklearn.model_selection import train_test_split
from   sklearn.preprocessing   import StandardScaler
from   sklearn.decomposition   import PCA
from   sklearn.metrics         import confusion_matrix
from   sklearn.linear_model    import LogisticRegression
from   sklearn.metrics         import confusion_matrix
from   sklearn                 import datasets

from   factor_analyzer         import FactorAnalyzer
from   factor_analyzer.factor_analyzer\
                          import calculate_bartlett_sphericity
from  factor_analyzer.factor_analyzer \
                          import calculate_kmo

from   matplotlib.colors       import ListedColormap
from   matplotlib.colors       import ListedColormap
from   matplotlib.colors       import ListedColormap
from   matplotlib.colors       import ListedColormap
from   csv                     import writer
from   csv                     import reader
from   copy                     import deepcopy
#================================
# This line is to change the working directory to where the best files are:
os.chdir("C:/Users/Pedram/Desktop/Spring 2022/Stat 795 Practicum in Statistics
Consulting/Climate Data/Best files")

# To confirm the change in the working directory:
cwd = os.getcwd(); print(cwd)

#=====================================

# In this section, we have cleaned the data. These files are red and stored
# as Pandas Data Frame. Each of the Data Frames is a attributes (property).
# Some of the files contained improper values and/or incomplete data points
# or (rows).
# Note : relhum20.csv contains cells with two different values (max and min)
# This file needs to be reprocessed to have two separate columns for each of
# the attributes (relhum_max and relhum_min) - to be worked on

df_avgsnf = pd.read_csv('avgsnf20.csv')
avgsnf    = pd.DataFrame(df_avgsnf)
avgsnf    = avgsnf.loc[(avgsnf["JAN"]!="T") & (avgsnf["FEB"]!="T") &
(avgsnf["MAR"]!="T") & (avgsnf["APR"]!="T") & (avgsnf["MAY"]!="T") &
(avgsnf["JUN"]!="T") & (avgsnf["JUL"]!="T") & (avgsnf["AUG"]!="T") &
(avgsnf["SEP"]!="T") & (avgsnf["OCT"]!="T") & (avgsnf["NOV"]!="T") &
(avgsnf["DEC"]!="T")]
avgsnf    = avgsnf.reset_index(drop = True)


df_hghtmp = pd.read_csv('hghtmp20.csv')
hghtmp    = pd.DataFrame(df_hghtmp)

df_lowtmp = pd.read_csv('lowtmp20.csv')
lowtmp    = pd.DataFrame(df_lowtmp)

df_nrmavg = pd.read_csv('nrmavg.csv')
nrmavg    = pd.DataFrame(df_nrmavg)

df_nrmcdd = pd.read_csv('nrmcdd.csv')
nrmcdd    = pd.DataFrame(df_nrmcdd)

df_nrmhdd = pd.read_csv('nrmhdd.csv')
nrmhdd    = pd.DataFrame(df_nrmhdd)

df_nrmmax = pd.read_csv('nrmmax.csv')
nrmmax    = pd.DataFrame(df_nrmmax)

df_nrmpcp = pd.read_csv('nrmpcp.csv')
nrmpcp    = pd.DataFrame(df_nrmpcp)

df_nrmsnw = pd.read_csv('nrmsnw.csv')
nrmsnw    = pd.DataFrame(df_nrmsnw)
nrmsnw    = nrmsnw.loc[(nrmsnw["JAN"]!="M") & (nrmsnw["FEB"]!="M") &
(nrmsnw["MAR"]!="M") & (nrmsnw["APR"]!="M") & (nrmsnw["MAY"]!="M") &
(nrmsnw["JUN"]!="M") & (nrmsnw["JUL"]!="M") & (nrmsnw["AUG"]!="M") &
(nrmsnw["SEP"]!="M") & (nrmsnw["OCT"]!="M") & (nrmsnw["NOV"]!="M") &
(nrmsnw["DEC"]!="M") & (nrmsnw["ANN"]!="M")]
nrmsnw    = nrmsnw.reset_index(drop = True)

df_pctpos = pd.read_csv('pctpos20.csv')
pctpos    = pd.DataFrame(df_pctpos)

df_prge   = pd.read_csv('prge0120.csv')
prge      = pd.DataFrame(df_prge)
```

```
df_wndmin = pd.read_csv('wndmin20.csv')
wndmin    = pd.DataFrame(df_wndmin)

df_wndmax = pd.read_csv('wndmax20.csv')
wndmax    = pd.DataFrame(df_wndmax)

df_hummin = pd.read_csv('relhum20min.csv')
hummin    = pd.DataFrame(df_hummin)
hummin    = hummin.loc[(hummin!=0).all(axis=1)]
hummin    = hummin.reset_index(drop = True)

df_hummax = pd.read_csv('relhum20max.csv')
hummax    = pd.DataFrame(df_hummax)
hummax    = hummax.loc[(hummax!=0).all(axis=1)]
hummax    = hummax.reset_index(drop = True)

#=========================================

# attribs is a list of all of attribute.
# Reminder: relhum20.csv is not included yet. It must be preprocessed as explained
# earlier in the attributes.

# The list of the NAMES of the attributes
attribs =
['avgsnf','hghtmp','lowtmp','nrmavg','nrmcdd','nrmhdd','nrmmax','nrmpcp','nrmsnw','pctpos','prge','wndmin', 'wndmax', 'hummin','hummax']

# The list of the Attributes themselves
Attribs = [avgsnf , hghtmp , lowtmp , nrmavg , nrmcdd , nrmhdd , nrmmax , nrmpcp ,
nrmsnw , pctpos , prge , wndmin, wndmax , hummin , hummax]

# ===========================================================

# n is the number of the attributes in the attribute list
# m and nx1vector values of which are the length of each attributes which are
# the complete cases in each attributes.

n     = len(attribs)
m     = np.zeros([n,1])
index = 0
l     = 1000

print(l)
```

```
#====================================================
# To find the csv file with the most restricted set of data:

for k in range(n):

    m[k]  = Attribs[k].shape[0]
    if l  > m[k]:
        l = np.linalg.norm(m[k]).astype(int)
        index = k

shortest = Attribs[index]
print(index)
print(l)
print(attribs[index])
print(n)
shortest

#====================================

#shortest
codetowns  = shortest.iloc[:,0:2]
codes      = np.array(codetowns.iloc[:,0],dtype=int) #codes
towns      = codetowns.iloc[:,1]  #towns
cityList   = deepcopy(shortest['TOWN'].tolist())
cityNump   = deepcopy(shortest['TOWN'].to_numpy())

numCities  = len(codes)
lenCodes   = len(codes)
lenAttribs = len(attribs)
i   = 0; k = 0; count1 = 0; count2 = 0
vec = np.zeros([numCities,12])
attribList = []  #Initilization of the attribute list for each city. Will be used towards
                 #indexing

Dataset    = np.zeros([numCities, lenAttribs])
numCities
np.shape(Dataset)

index_values  = attribs
column_values =
['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC','AVE']
summaryList   = ['mean', 'variance', 'std', '1st quartile', 'median', '3rd quartile', '85
percentile']
summaryLength = len(summaryList)
```

```
summary       = np.zeros([summaryLength, lenAttribs])

index_values  = attribs
column_values =
['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC','AVE']


for i in range(lenCodes):
    cityNump[i] = np.zeros([lenAttribs,13])
    code        = codes[i]
    town        = towns[i]  # redundant? Not yet used

    attribList = []  #Initilization of the attribute list for each city. Will be used towards
                    #indexing
    for k in range(lenAttribs):

        if (code in np.array(Attribs[k].iloc[:,0])):

            ind                 = np.where(Attribs[k].iloc[:,0] == code)[0][0]
            cityNump[i][k,0:12] = np.array(Attribs[k].loc[ind,"JAN":"DEC":1],dtype = float)
            cityNump[i][k,12]   = round(np.average(cityNump[i][k,:]),2)
            count1              = count1 + 1

        else:
            cityNump[i][k,:]  = -10000
            count2    = count2 +1

    df = pd.DataFrame(data = cityNump[i],  index = index_values, columns =
column_values)

    Dataset[i,:]  = np.array(df.iloc[:,12],dtype = float)


    df.drop(df.index[df['JAN'] == -10000], inplace=True)


    vars()[cityList[i]] = df
    df.to_csv("vars()[cityList[i]].csv")
    address = 'C:/Users/Pedram/Desktop/Good Files/'+str(cityList[i])+'.csv'
    df.to_csv(address)
```

```
Datasetdf     = pd.DataFrame(data = Dataset,  index = cityList, columns = attribs)

for i in range(len(attribs)):
   Datasetdf.drop(Datasetdf[Datasetdf[attribs[i]]==-10000].index, inplace = True)

#mean = np.mean(np.array(Datasetdf.iloc[:,12],dtype = float))

Datasetdf.to_csv("mainData")
address = 'C:/Users/Pedram/Desktop/Good Files/mainData.csv'
Datasetdf.to_csv(address)


Datasetdf

#===================================================

###############
# The PCA section
###############

#allData   = np.array(Datasetdf.loc[:,:])
allData    = Datasetdf.to_numpy()
print(type(allData))
print(np.shape(allData))
allData

X        = np.copy(allData)
X        = preprocessing.StandardScaler().fit(X).transform(X)
np.shape(Datasetdf)

model     = PCA()
results   = model.fit(X)
Z        = results.transform(X)
XX       = model.fit_transform(X)

sigmaSqrd = results.explained_variance_
sigmaSize = len(sigmaSqrd)
np.size(model.explained_variance_ratio_)

# Check if the result is zero-matrix ( ,>  < )
XX-Z

#=================================================
pyl.plot(sigmaSqrd)

plt.xlabel("Percentage Variance Explained")
plt.ylabel("Number of Components")
#plt.grid()
plt.show()

#=================================================

sigmaSqrd = np.copy(results.explained_variance_) #The Variances along the Principal Components
sigmaSize = len(sigmaSqrd)

indeces = np.arange(1,sigmaSize+1) # The 1D array from 1 to size of variances
indexes = np.copy(indeces)
indeces = np.round_(np.reshape(indeces,(sigmaSize,1)),decimals = 0)


sumVars = np.sum(sigmaSqrd)
print("Sum of the Variances is:",round(sumVars,3),"\n")

expVars = np.zeros([sigmaSize,1]) # Initiation of "Explained Variances"
sumexVa = np.round_(np.sum(sigmaSqrd*100/sumVars),decimals = 3)
print("The sum of the proportions of the Variances is:", sumexVa)
print("\n")


np.set_printoptions(suppress=True)

expVar   = np.round_(model.explained_variance_ratio_*100,decimals=2)
expVar   = np.reshape(expVar,(sigmaSize,1))

sumVars  = np.cumsum(100/sumVars*sigmaSqrd).round(2)
sumVars  = np.reshape(sumVars,(sigmaSize,1))

pcaVars  = np.concatenate([indeces,expVar],axis = 1)
pcaVars  = np.concatenate([pcaVars,sumVars],axis = 1)


print("The shape of indeces is    ",np.shape(indeces))
print("The shape of the expVars is",np.shape(expVars))
print("The shape of the sumVars is",np.shape(sumVars))
print("\n")
print("The shape of the pcaVars is",np.shape(pcaVars))
print("\n")
print(pcaVars)

pcaHeaders = ['PCs','Variance Proportions %', 'Cummulative Vars Proportions']
df = pd.DataFrame(data = pcaVars,  index = indexes, columns = pcaHeaders)
df['PCs']  = df['PCs'].astype(int)
Df

# ======================

pcLabels = []
type(pcLabels)
for i in range(sigmaSize):
   labels = "PC " +str(i+1)
   pcLabels.append(labels)
pcLabels

#=======================

# The coordinates of the data set along with the principal axises
pcDataFrame = pd.DataFrame(data = np.round_(Z,decimals = 2) , index =
Datasetdf.index, columns = pcLabels)
pcDataFrame

#========================

df = Datasetdf
n_components = 6

pca = PCA(n_components = n_components)
components = pca.fit_transform(df)

total_var = pca.explained_variance_ratio_.sum() * 100

labels = {str(i): f"PC {i+1}" for i in range(n_components)}
#labels['color'] = 'Median Price'

fig = px.scatter_matrix(
   components,
   #color=boston.target,
   dimensions=range(n_components),
   labels=labels,
   title=f'Total Explained Variance: {total_var:.2f}%',
)
fig.update_traces(diagonal_visible=False)
fig.show()

df = Datasetdf
X = df[['hghtmp','wndmax','nrmpcp' ]]

pca = PCA(n_components=3)
components = pca.fit_transform(X)

total_var = pca.explained_variance_ratio_.sum() * 100

fig = px.scatter_3d(
    components, x=0, y=1, z=2,
    title=f'Total Explained Variance: {total_var:.2f}%',
    labels={'0': 'PC 1', '1': 'PC 2', '2': 'PC 3'}
)
fig.show()

# ==============================================

# ==================
# Factor Analysis starts here
# ==================

chi_square_value,p_value = calculate_bartlett_sphericity(df)
(chi_square_value,p_value)

print("The Chi-Squared Value is", chi_square_value)
print("The p_value is",p_value)

# In this Bartlett ’s test, the p-value is 0. The test was statistically significant,
# indicating that the observed correlation matrix is not an identity matrix.

# ==========================================
kmo_all,kmo_model = calculate_kmo(df)
print("Tha KMO value is",kmo_model.round(4))

#==========================================

# Create factor analysis object and perform factor analysis
fa = FactorAnalyzer()
#fa.analyze(df, 25, rotation = None)
# Check Eigenvalues
```

```
#ev, v = fa.get_eigenvalues()

fa.fit(df)
eigen_values, vectors = fa.get_eigenvalues()

eigen_values   = np.reshape(eigen_values,(len(eigen_values),1))

factorEigVal   = np.concatenate([indeces,eigen_values],axis = 1)

factorEigValDf = pd.DataFrame(data = factorEigVal, index = np.arange(1,len(attribs)+1),
columns = ['Factor Number', 'Eigen Value'])

factorEigValDf['Factor Number']  = factorEigValDf['Factor Number'].astype(int)
factorEigValDf

# ================================================

# Create scree plot using matplotlib
plt.scatter(range(1,df.shape[1]+1),eigen_values)
plt.plot(range(1,df.shape[1]+1),eigen_values)
plt.title('Scree Plot')
plt.xlabel('Factors')
plt.ylabel('Eigenvalue')
plt.grid()
plt.show()

# ===============================================

n           = len(factorEigValDf[factorEigValDf['Eigen Value']>1])

factorsList  =[]
for i in range(n):

    factorNum = 'Factor '+str(i+1)
    factorsList.append(factorNum)



fa          = FactorAnalyzer()
fa.set_params(n_factors = n , rotation = "varimax")
fa.fit(df)
loadings     = fa.loadings_
varimaxRot   = np.round_(loadings,decimals = 4)
varimaxRotDf = pd.DataFrame(data = varimaxRot, index = attribs, columns = factorsList)
```

```
varimaxRotDf

# ==============================================

factorVars     = fa.get_factor_variance()
factorsVarsInx = ['SS Loadings', 'Proportion Var', 'Cumulative Var']
factorVarsDf   = pd.DataFrame(data = factorVars , index = factorsVarsInx, columns =
factorsList)
factorVarsDf

# ===============================================

#################################
# The Linear Regression Analysis section
#################################


# To set up the attribute to be the response variable and the rest of the 14 attributes to
be
# the independend variables.

Response  = 'hghtmp'

xdf       = Datasetdf.loc[:, Datasetdf.columns!= Response]
ydf       = Datasetdf.loc[:,Datasetdf.columns == Response]

#========================================


xdf = sm.add_constant(xdf)

model = sm.OLS(ydf, xdf).fit()
predictions = model.predict(xdf)

model.summary()

# ========================================

#xdf = xdf[['nrmavg','nrmcdd','nrmhdd','nrmmax','nrmmax','prge']]
xdf       = Datasetdf.loc[:, Datasetdf.columns!= Response]
xdf = xdf.loc[:, ['nrmavg','nrmcdd','nrmhdd','nrmmax','nrmmax','prge']]
type(xdf)

#==========================================
xdf = sm.add_constant(xdf)

model = sm.OLS(ydf, xdf).fit()
predictions = model.predict(xdf)

model.summary()

# ========================================

df = pd.read_csv("outlier_removed.csv")
print(np.shape(df))
df.head()

# ========================================

# To set one of the components as the response variable:

Response  = 'hghtmp'

xdf       = df.loc[:, df.columns!= Response]
ydf       = df.loc[:,df.columns == Response]

# ============================================

xdf = sm.add_constant(xdf)

model = sm.OLS(ydf, xdf).fit()
predictions = model.predict(xdf)

model.summary()

# =============================================
```