**Final Project  
Prabesh Rai**

**Dan Horner   
 Intro to Data Science (DS 210)**

**Introduction and Stating the Question  
10 pts**

In this project we are going to use our knowledge and skills to perform Exploratory Data Analysis and visualization. We will be using the data from GLOBE Scientists’ Blog. Their data contains 6 columns and 55 rows but for our project we will only use two columns “Chirps 15s” and “TempFahrenheit” to perform our analysis. Over the course of this analysis, we will progress through the data science life cycle in our attempt to answer if the outside temperature can be estimated by the frequency of cricket chirps. We will achieve our project goal by using python’s panda’s library for data cleaning and initial exploration and use KNIME analytics program to create predictive model for insightful visualization.

Data Source link: [Measuring temperature using crickets | GLOBE Scientists' Blog](https://www.globe.gov/explore-science/scientists-blog/archived-posts/sciblog/index.html_p=45.html)

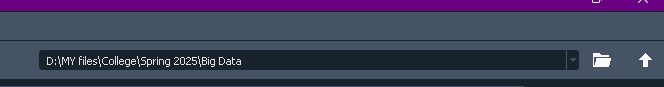
**Exploratory Data Analysis  
40 pts**

The crucial step in data science life cycle is Exploratory Data Analysis but before we jump into analyzing the relationship within our data, we need to ensure the data is clean and ready for analysis. We will use python’s panda’s library to clean our data and find any issues such as outliers or missing values.

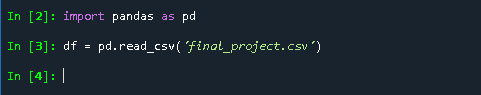
**Data Collecting and Importing Steps:**

Using our data source link above let’s pull the data and create a CSV file that we can read and clean using python pandas. We can name the according to our needs. In my case I will name it as “final\_project” and save it in one of my working directories.

Then I will open up my Spyder IDE and change my working directory where I have saved that “final\_project.csv” file. You can find this option in the top right corner of the Spyder program.



Now we can use our spyder console to import pandas and setup our data frame to read the csv file using the steps blow. The df stands for data frame that we working with and “read\_csv” is the function that reads the csv file.



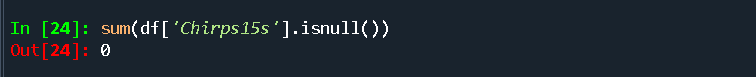
Before we move on to cleaning process lets set up our file to display maximum column instead of default by using



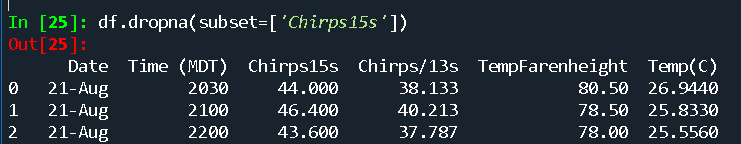
Before we start cleaning our data lets analyze the columns that we need for our analysis and check for invalid or outliers value and missing values.

For the **Chirps15s feature**, do the following:

* Let’s us check if there are any null values in our Chirps15s column using:: **sum(df['Chirps15s'].isnull())**

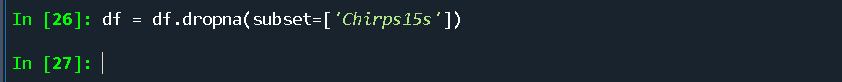


* We got out as zero which means there are no null values in our column.
* If we had any null values in our columns than we can remove them using:: df.dropna(subset=['Chirps15s'])



* That will remove all the null values from the column and display all the columns but won’t update it in our data frame. To update our data frame with that or any changes we need to assign it to our data frame using

**df = df.dropna(subset=['Chirps15s'])**



* Now that we have taken care of the null values lets look into distinct value counts using **df['Chirps15s'].value\_counts()**
* From the output result we can easily see how many times each value has repeated and if there are any invalid or outliers value.
* For example, we can identify values that are extremely low or high with less frequency for numerical values and for categorical values if the column only supposed to contain “Male” and “Female” and we see distinct “M” or “F” or in other format.
* This will give us a starting point for our data cleaning process. But in our case, we have numerical values and need to consider if the values are outliers or invalid.
* Before jumping into conclusion lets look into our minimum and maximum value in our column using

**df['Chirps15s'].min()**

**df['Chirps15s'].max()**



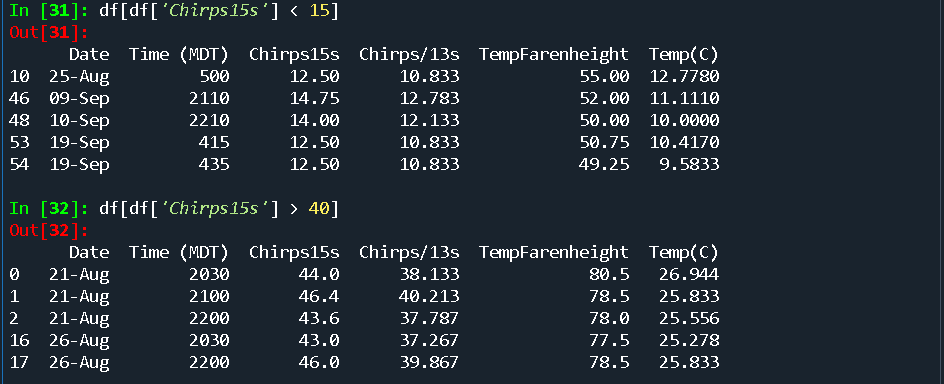
* Looking at the maximum and minimum value and their frequency they can be a potential outliers or invalid values in our data set. The frequency of 46.400 is one and 12.500 is three, it is hard to tell that they are actually an outlier in our data without visual representation and further observation and its relation with temperature.

  df['Chirps15s'].value\_counts()

* Based on the information from our source link it reports that around 49 degrees Fahrenheit they did not heard any cricket chirps at all making any values below 50 degrees Fahrenheit our potential outliers and not a significant on the high temperatures mentioned but we can at least assume that the higher the temperature the more inaccurate values are recorded.
* But if we were sure they were potential outliers we can easily check how many values are in our data using

**df[df['Chirps15s'] < 15] assuming values below 15 are outliers**

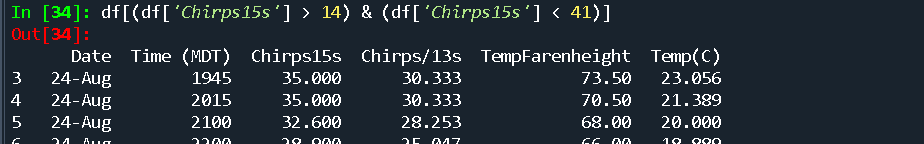
**df[df['Chirps15s'] > 40] assuming values above 40 are outliers**



* Now we can redefine our data between the range we want and update our df using

**df[(df['Chirps15s'] > 14) & (df['Chirps15s'] < 41)]**

It will display all the values between those range



Still, we are assuming those range and if we need to update, we just assign the filtered column of Chirps15s to our data frame using

**df = df[(df['Chirps15s'] > 14) & (df['Chirps15s'] < 41)]**

For our case I will not update the data frame with this range because I still need to check the temperature column and its relation with our Chirps15s column.

Now that our Chirps15s column is cleaned we can move to our temperature column and repeat the same process.

For the **TempFahrenheit feature**, do the same as you did for Chirps15s:

* First, we will display and analyze all the distinct values in our TempFahrenheit using

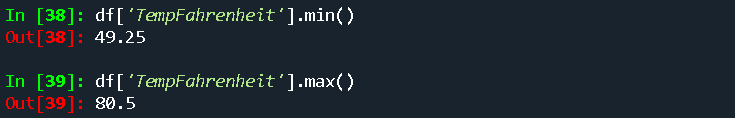
**df['TempFahrenheight'].value\_counts()**



* It will give us all the distinct values and its count. From the source it was mentioned that below 50-degree Fahrenheit the reports were not clear and assuming higher the temperature higher inconsistency in the data collection.
* Let’s examine the minimum and maximum values of this column first using

**df['TempFahrenheit'].min()**

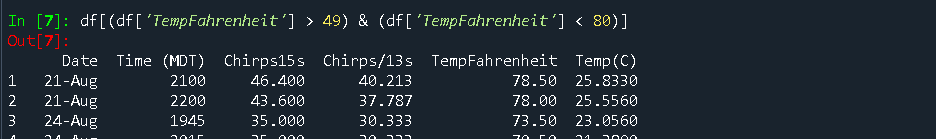
**df['TempFahrenheit'].max()**



* We can assume that any values below 50-degree Fahrenheit temperature might be outliers and potentially higher will do the same but just looking at the data we cannot assume and we need to do further analysis.
* If we were sure they were invalid or outliers values then we can easily remove them or sort our values from our column using

**df[(df['TempFahrenheit'] > 49) & (df['TempFahrenheit'] < 80)]**

and assign this to our data frame. But we will not perform this step because we want all the values for further analysis.

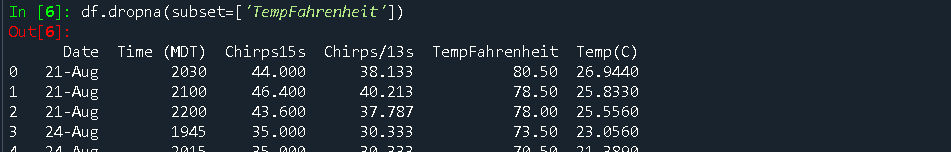


* To ensure there are not null values we will remove all the rows that contains null values using

**df.dropna(subset=['TempFahrenheit'])** and assign to data frame to update our data using

**df = df.dropna(subset=['TempFahrenheit'])**

For our case we will keep the data as it is because there are no null values in our column.



Now let us update our data frame to only two columns that we need for this project and they are **‘Chirps15s’** and **‘TempFahrenheit’** using::

**df = df[['Chirps15s', 'TempFahrenheit']]**



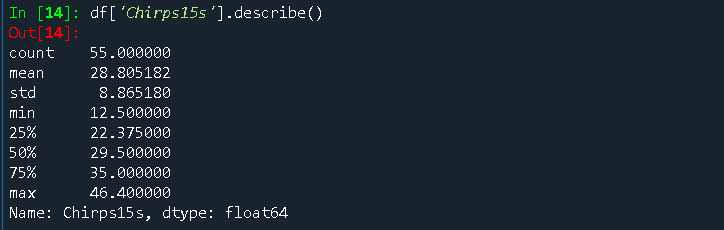
* Now we can find our columns mean and median values using::

**df.describe()**



* Beside mean and median we can also see our maximum and minimum values for each column. We can also find these values for individual columns using::

**df['Chirps15s'].describe()**



* We can also use::

**df[‘Chirps15s’].mean()** or

**df['Chirps15s'].median()**

To find only the mean or median value for individual columns. For this project we will continue with

**df.describe()**

for efficiency and the 50% values are the median values for each column.

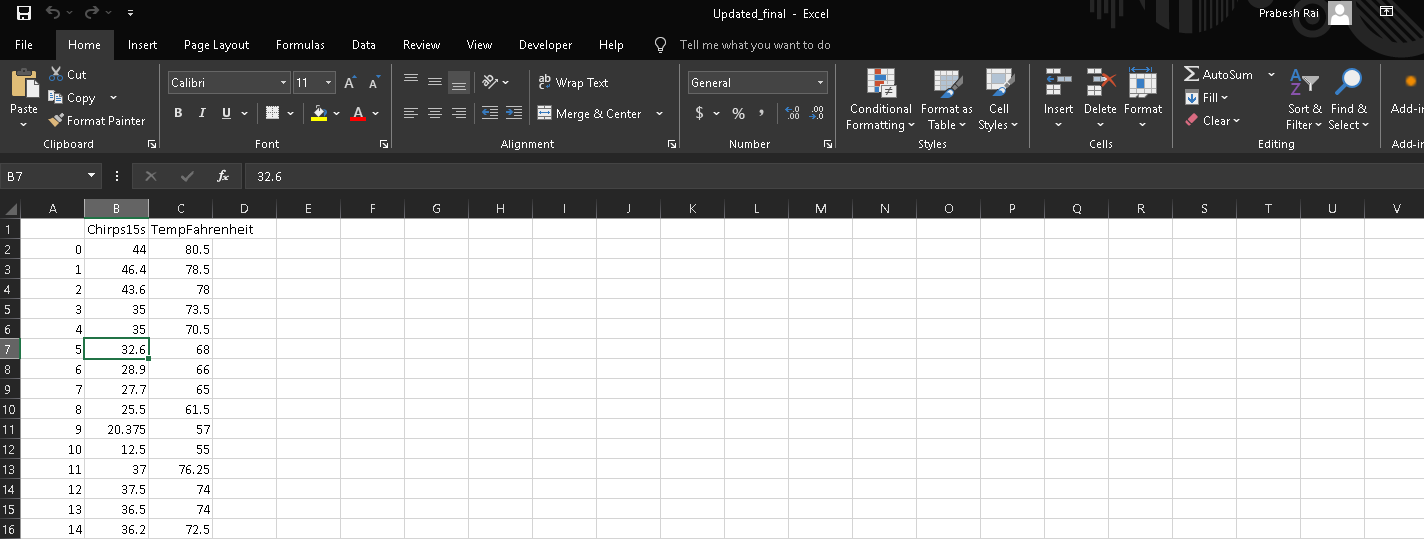


* Now we can see our mean first quartile and third quartile values as well. Which will help us identify our outliers.
* Let’s look at the Chirps15s columns value of 25% which is our lowest interquartile range and 75% which is our highest interquartile range.
* We have 22.38 rounded for the least and 35.00 for the maximum range any values outside of this range are our outliers.
* Same with the column TempFahrenheit we have 60.50 as the least and 71.50 as the maximum range any values outside of this range are our outliers.
* For this project we will not remove any of those outliers because we want to visualize them in our analysis step.
* Now let’s save a new csv file with the updated columns only before we move forward, we can create new csv file using

df.to\_csv('Updated\_final.csv')



* Next double check you folder to see if the file is saved with the columns we wanted.



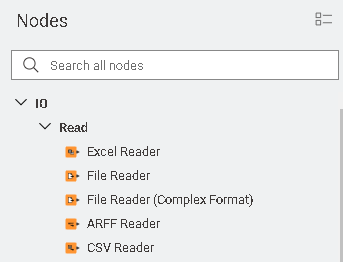
Now we can import our new file in our KNIME analytics platform to create a scatterplot to analyze the relationship between these two quantitative variable columns.

**Scatterplot Workflow in KNIME:**

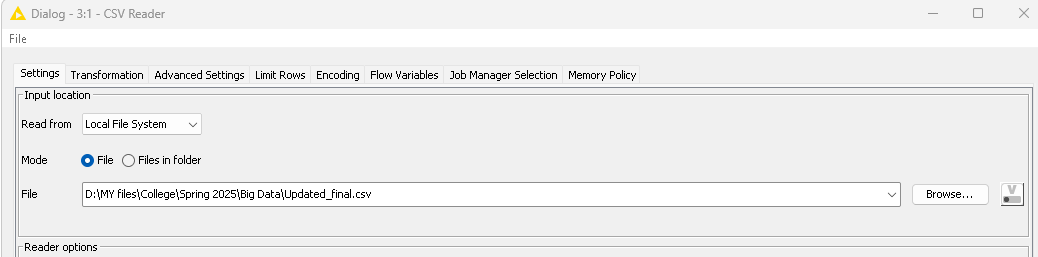
* We could have also done the same Data cleaning or Exploratory Analysis using KINME from the beginning. Moving further as building our Scatterplot I will explain those steps as well.
* Let us import and read our new data file using KNIME. I will create new project and name it ‘final\_project’.
* Using Nodes I will import CSV Reader. I can directly type the node name in the search box or follow the step:

Goto

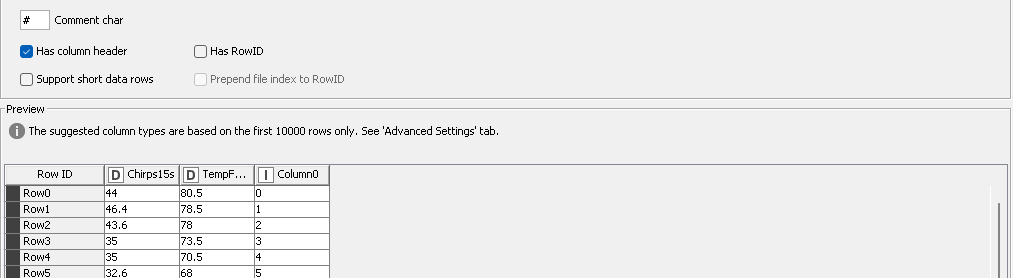
IO 🡺 Read 🡺 CSV Reader 🡺 drag and drop or double click to add on your project.



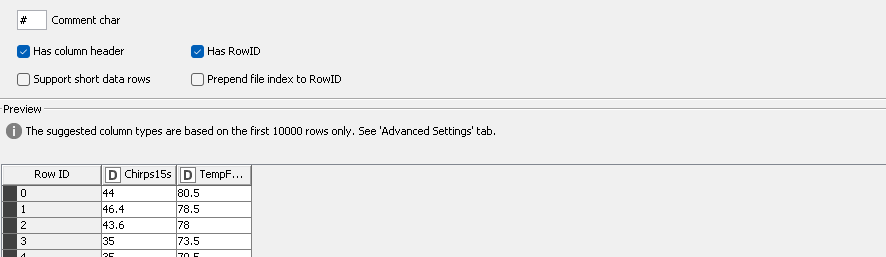
* Then right click on the CSV Reader node that you have added in your project and choose Configure.
* A configure dialog box will appear where you can choose your file using the Browse option and select it.



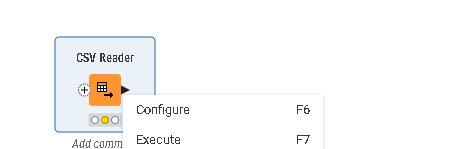
* On the preview section you will see your Row ID and columns names with their values.
* Since we created our file with its own columns name and RowID we will check on ‘Has column header’ and ‘Has RowID’ to import our file data in original format.



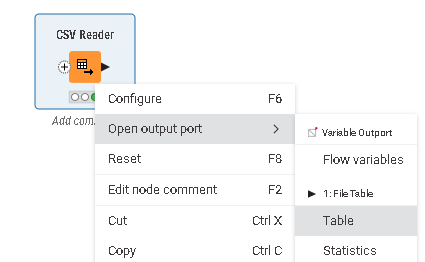
* After we check those options, it will remove any unnecessary columns and headers.



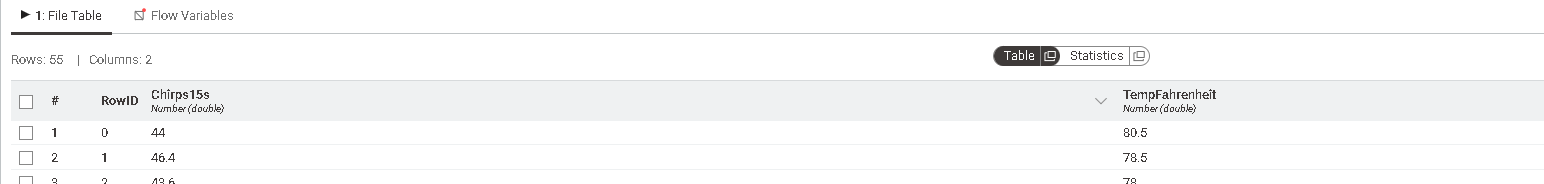
* Now click on Apply or OK to continue. And again, right click on the CSV Reader node and click on Execute.

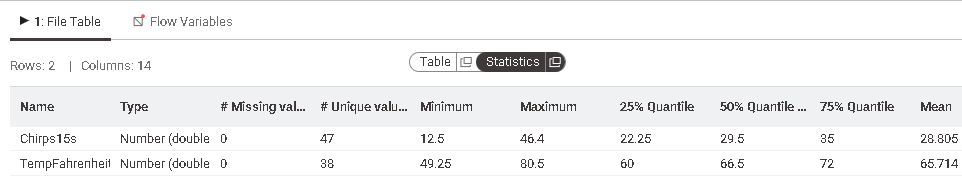


* Next, we can right click on the node and click Open output port and Table or Statistics to view the file data.



* The Statistics table will give us the same values as df.describe() in python pandas. Or we can use the preview box on the bottom of our project to view the same data. If we click Table, it will display all the rows and columns and Statistics will display all the columns mean, median, maximum, Interquartile range and Standard Deviation and so on.

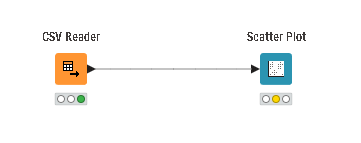




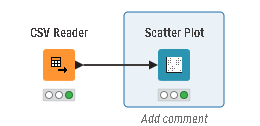
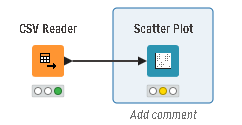
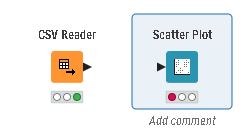
* Since we have already identified all of these values in our previous Exploratory Data analysis steps, we will move forward to create scatterplot and analyze the relation between our two columns.
* Again, we will go to our Nodes

Views ► Visualization Column Appender 🡺 Scatter Plot 🡺 and insert it using drag and drop or double click

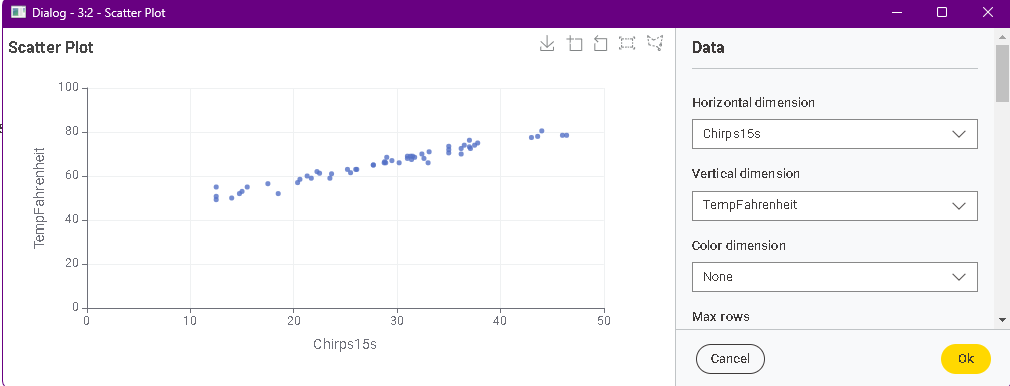
* Now connect the CSV Reader and Scatter plot use the arrow from CSV Reader and connect it to Scatter Plot.



* The three dots on the bottom of the nodes are status of your nodes it will be red, yellow and green. If it is red than we need to configure it and yellow means not executed and green means executed.



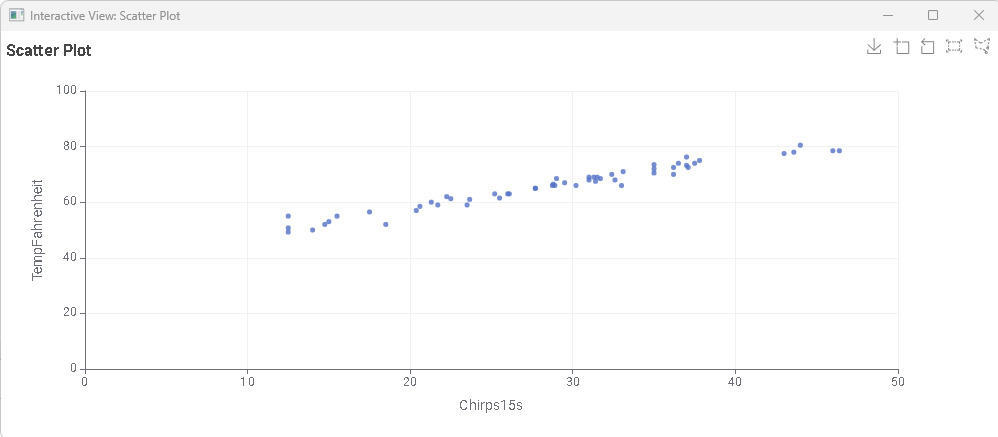
* Now we need to configure our Scatter Plot’s X and Y axis for our visualization. Right click on the scatter plot node and click on configure.
* A dialog box appears and for our project I will choose Chirps15s column in Horizontal dimension x-axis and TempFahrenheit in Vertical dimension y-axis.



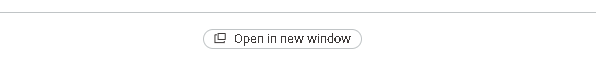
* And click on ok and execute it. After that you can see the scatter plot in the preview box below or you can

Right click Scatter plot node 🡺 Open view 🡺 it will open the interactive view of Scatter plot.

Note: We can only Zoom in or out when we interact with the scatter plot.



* Or you can Click on Open in new window option is the preview section to display the above result.



* Looking at the scatter plot we can conclude that there is a positive relationship between the number of chirps in 15 seconds and the temperature in Fahrenheit. We can see that as the temperature goes up the number of chirps also increases.

**Refining the Question  
10 pts**

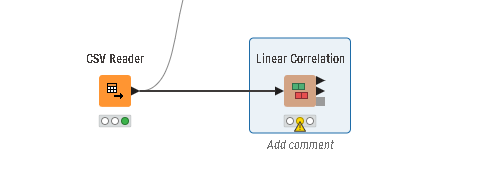
Now lets us look back into our exploratory data analysis and examine our findings and determine if we need to change our original question “Can the outside temperature be estimated by the frequency of cricket chirps?” Based on our current findings, the raw data, its ranges, potential outliers and visual representation in our scatterplot there is a relationship that exist relevant to the question. If the raw data had missing values, and scatterplot had shown no pattern or non-linear relationship we could have conclude that this specific data is not sufficient to answer our question. Therefore, we will proceed with our question using the current data because we have a visible positive linear relationship between “Chirps15s” and “TempFahrenheit” data.

**Model Building   
  
30 pts**

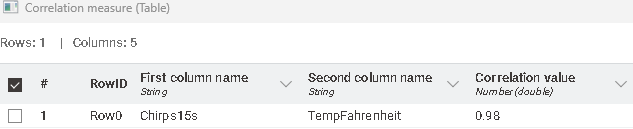
Now that we have reexplored our data and confirmed a positive linear relationship exist, in our next data science process we will build a predictive model using KNIME. This step will use supervised learning because we will use our current dataset to train the model than it will be able to predict temperature based on unknown Chirps count that we provide.

Predictive model Workflow in KNIME:

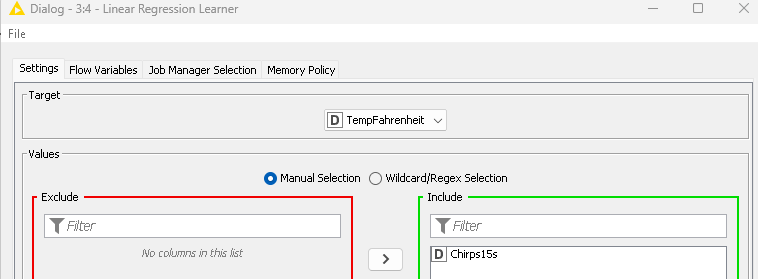
* From our Nodes we will import Linear Correlation node in out project file. In our case “final\_project” and connect it with our CSV Reader node.



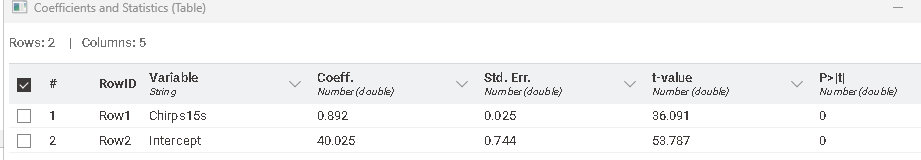
* Right Click ► Configure ► Dialog box appears ► Include columns in our case “Chirps15s” and “TempFahrenheit” ► Click Apply and OK ► Execute
* It will pull out our Correlation value. The value has to be close to positive one to hold strong positive relation and in our case, we got “0.98” which is strongly close to positive one.



* Next, we will add another node called Linear Regression Learner and connect it with our CSV Reader node.
* Then we will configure it as the target variable will be TempFahrenheit and Included Values as Chirps15s



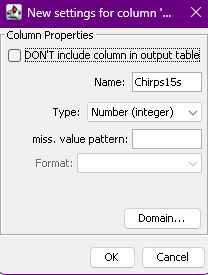
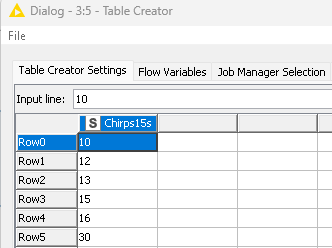
* Next, we will click on Apply and OK and execute it. We will be able to see table like one below



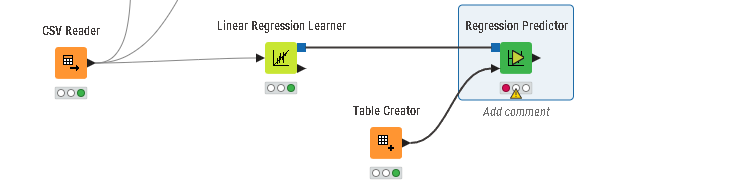
* Under the Coeff. Column the Chrips15s value is our m value and Intercept is our b value. The Chirps15s value is our slope known as “m” and intercept value is our y-intercept value known as “b” in our linear equation which is:

y = mx + b

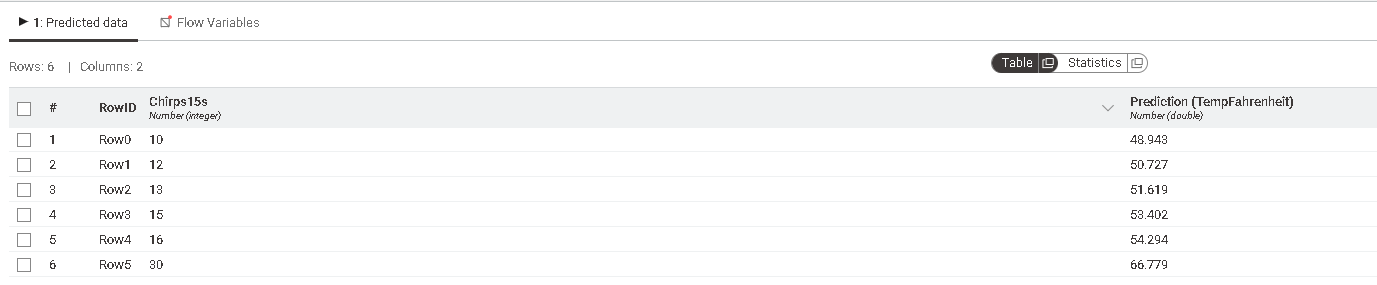
* The value “x” will be the one that we provide as Chirps count to the model which will help us find the “y” value that is TempFahrenheit.
* Now our linear equation that will predict the TempFahrenheit value will be ► y = 0.983 x + 40.025
* We can also call this as our line of best fit to make a prediction for the TempFahrenheit value.
* Now we will predict few values to test our model. We can create a new CSV file with the same column name as “Chirps15s” and assign some random values to predict “TempFahrenheit” values or we can add a Table Creator node in KNIME and assign the same column name and values that we want to provide.
* Search Table Creator in Nodes ► Right click Configure ► Left Double click to open column Properties ► Change the column name, Type as Number(integer) and click ok ► Click on the row and input values hit enter to keep adding row values for example in ours we have 10,12,13,15,16,30

* If we create new CSV file than we have to add another CSV reader and configure it before we can predict values.
* Next, we will add Regression Predictor Node from our Nodes and connect it with Linear Regression Learner and the new CSV file or in our case The Table Creator to predict the values.



* Next, we will just need to Configure our Regression Predictor and it will output the predicted values of TemFahrenheit with the Chirps15s values that we provided.



* Next, we will use our question and manually predict the outside temperature on a night when we hear 40 cricket chirps in 15 seconds.
* We will use our previous line of best fit equation ► y = 0.983 x + 40.025 and calculate the value of y step by step.
* Since the input is 40 which is our “x” value we will replace “x” using 40

y = 0.983 (40) + 40.025

y = 39.32 + 40.025

:. y = 79.345 or 79.35 degrees Fahrenheit (rounded)

**Interpretation/Summary  
10 pts**

Finally, in our project we used data science life cycle to determine if outside temperature can be estimated by cricket chirp frequency. We used data collection method to gather data from the source and perform ETL process to clean the data using pandas. We also used exploratory data analysis technique including scatterplot visualization to analyze the relationship between Chirps count 15 seconds and the temperature in Fahrenheit values. Which revealed a strong positive linear relationship as correlation coefficient of 0.98. Which confirmed that the data is suitable to build our supervised linear regression model which can be used to predict the estimated temperature based on cricket chirps count. With the help of our model, we predicted few temperature values for chirps counts and further we manually calculated the temperature based on 40 chirps count using our linear equation.