Final Project

Robert Daniels
Intro to Data Science (DS 210)

Introduction and Stating the Question
10 pts

Can we predict the outside temperature just by listening to the crickets? In this paper, we'll be progressing through the data science life cycle to determine a possible answer to this question. The data analysis cycle follows closely with the standard scientific method: form a hypothesis, determine an experiment, run that experiment / gather data, analyze said data, and then draw conclusions in relation to the hypothesis. The experiment and data collection phase has already been completed when we step into this process for data analysis.

The data set we will be working with is a 2-dimension frame representing measurements of ambient temperature and the related number of cricket chirps (as measured over a 15 second duration.) We load this file that is stored in .csv format into a dataframe made available through the Python PANDAS library. This library is an extension of the array functionality offered by the numpy package. After all, what is a matrix except arrays stacked next to each other?

The raw set contains 58 records (temperature :: chirps), with a null value found in 2 of these records. The presence of these null values suggest that we will need to transform and clean these data before running statistics or building a model. However, it is always a good idea to reduce any excess noise in the data set to yield the best possible result. The best possible result, in this case, means getting the best reading on an answer, as opposed to any one conclusion we may or may not wish to draw. We let the data drive the conclusion, not the other way around!

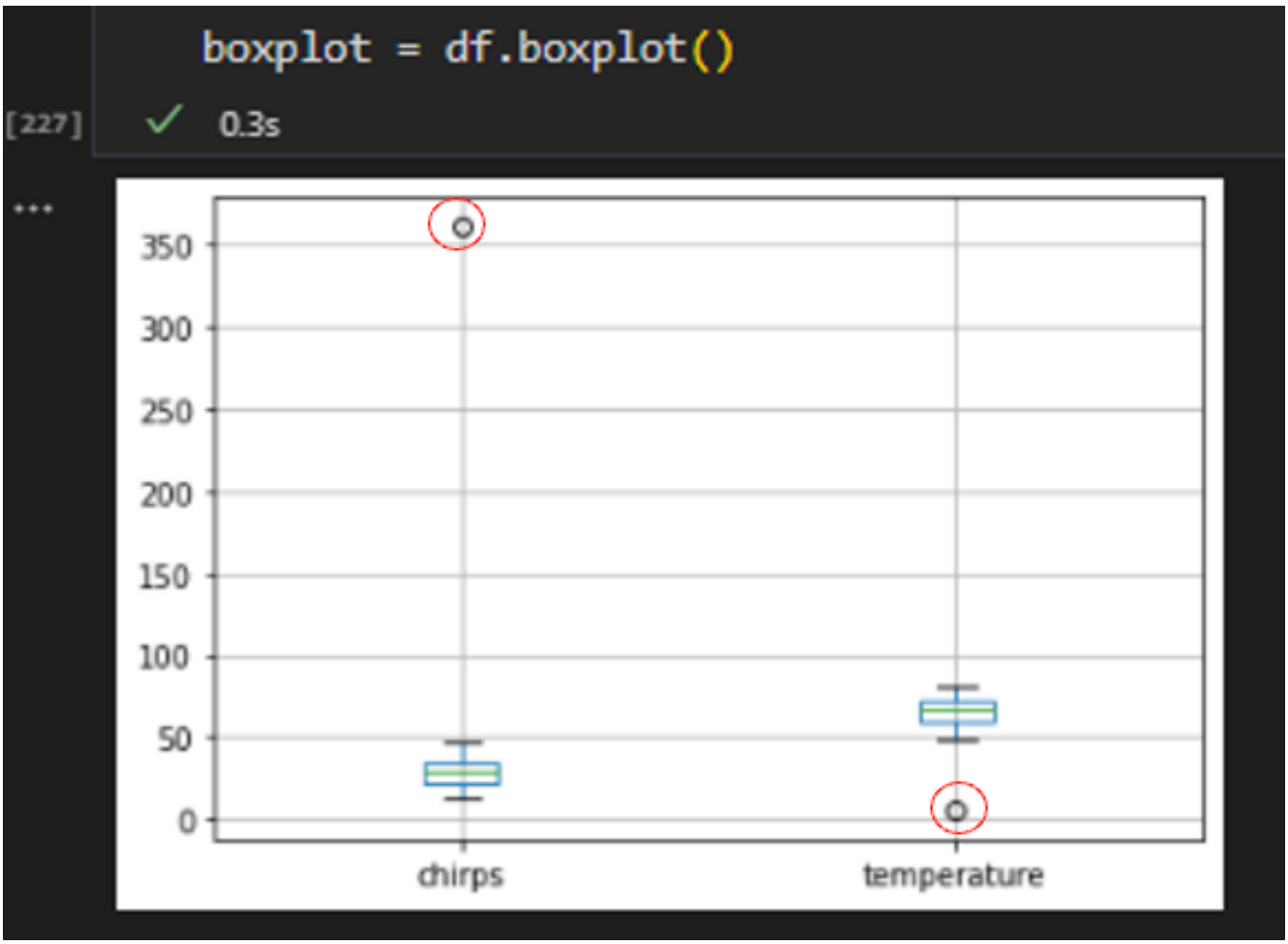
Exploratory Data Analysis
40 pts

We've formed our question and gotten some top-level ideas about our data set. Now it's time to get into exploratory analysis. Exploratory analysis in this case involves the initial immediate cleaning necessary regarding null values and outliers We cannot perform end product analysis on "bad" data. (The "Garbage In, Garbage Out" principle applies here!)

Right off the bat, we should figure out what to do with those null values. As there are only 2 records out of 58 that are missing critical data, we can safely remove these two records from the data set without losing an intolerable amount of material. To do this, we will use the PANDAS method .dropna(). The official full expression is df = df.dropna() which operates on the data frame, allowing null values to fall off the data set. Once called, this expression will remove records in either of the features that have null values.

To visualize any outliers in the features, we can use a tool in statistics called the box plot. A box plot conveys a visual representation of several key statistical measurements: the quartiles, the inner quartile range(IQR = Q3 – Q1), and the max / min value of the data set (outliers not included.) The outliers are classified as any data point above the third quartile + 1.5(Q3 – Q1) and any data point below the first quartile – 1.5(Q3 – Q1). This is thankfully something a computer can do quite well, once given the definition of the ranges. Thankfully, this functionality has already been built into the PANDAS library which allows us to simply render a plot with boxplot = df.boxplot() in a Jupyter notebook.

As can be seen below, there are outliers in these data. As these data are, statistically, not in line with the norm, it would not do to leave these in a data analysis attempting to build a model of usual behavior.



There exist numerous ways to “mask out” these outliers from the data fame. Below is the method that I took to do so. Each individual feature was spliced and copied into its own array (transposed from vertical to horizontal.)

Calling the .quantile() method in numpy, I was able to establish values for the relevant statistics needed to find outliers. Once ran, the outliers were returned to the console.

```
chirpsArray = df['chirps'].to_numpy()
```
[229] ✓ 0.6s Python

```
tempArray = df['temperature'].to_numpy()
```
[230] ✓ 0.6s Python

```
outliers =[]
noOutliers = []

def findOutliers(array):
    Q3 = np.quantile(array, 0.75)
    Q1 = np.quantile(array, 0.25)
    IQR = Q3 - Q1

    bottom = Q1 - (1.5 * IQR)
    top = Q3 + (1.5 * IQR)


    print("This is the bottom: ", bottom)
    print("This is the top: ", top)
    print("Q3: ", Q3)
    print("Q1: ", Q1)
    print("IQR: ", IQR)

    for i in array:
        if (i > bottom) & (i < top):
            noOutliers.append(i)
        else:
            outliers.append(i)

    print("The outliers, if any, were: ", outliers)
```
[231] ✓ 0.9s Python

find chirp outliers

```
findOutliers(chirpsArray)
```
[232] ✓ 0.1s Python

```
This is the bottom:  3.75
This is the top:  53.75
Q3:  35.0
Q1:  22.5
IQR:  12.5
The outliers, if any, were:  [361.0]
```

Once the outliers were known, they were reasonably easy to mask out so the analysis could continue.

remove outliers

```
df = df[df.chirps < 361]
df = df[df.temperature > 6]
```

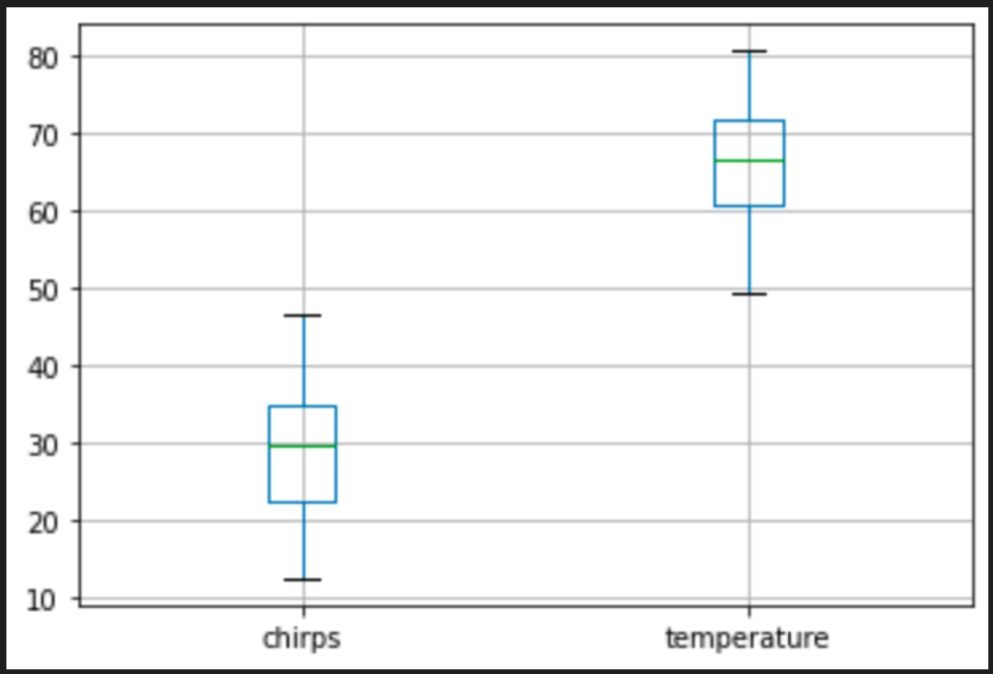
[15] ✓ 0.1s

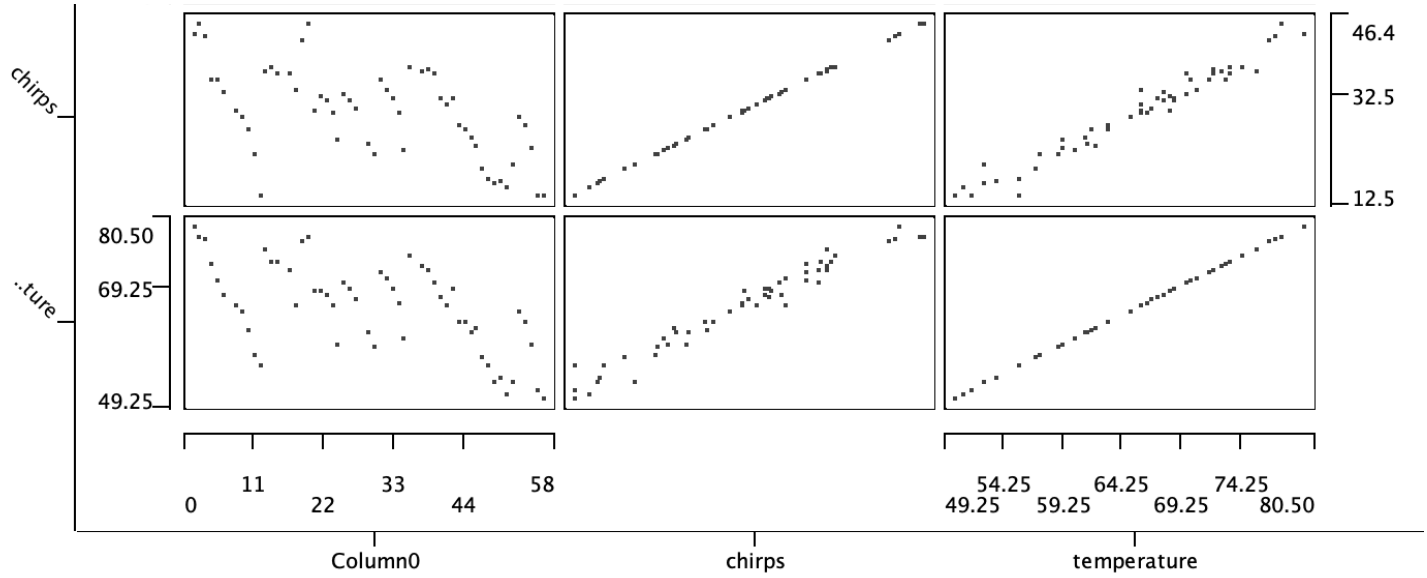
With the outliers removed, and the null values taken care of, we can now visualize and revisit our hypothesis.

```
boxplot = df.boxplot()
```

[17] ✓ 0.7s

...





Like the look of the graph above? This was generated by importing the data set into the KNIME platform. KNIME is a data analysis product that comes pre-loaded with a large toolbox of applications that can help our process along!

Refining the Question

10 pts

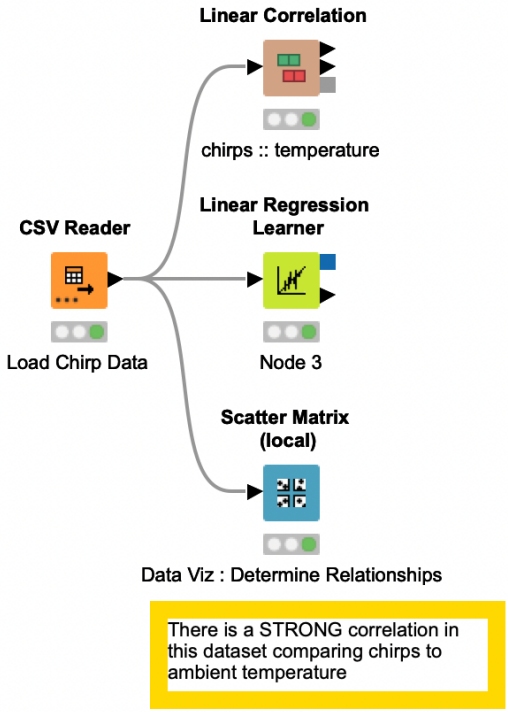
Now that we have a clean data set, we should refine our question to ensure we're still using our resources correctly. Our original hypothesis is that temperature CAN be predicted by chirp count. The null hypothesis is that we CANNOT predict such. While we won't be getting into statistical significance, eventually we would want to produce a finding with a p value less than or equal to .05 in order to confirm the original question. Looking at the scatter matrix above, visually there appears to be a striking linear relationship between these two variables. Put another way, there appears to be a strong correlation between temperature and chirp frequency. As such, we do not need to refine our original question currently. If this relationship did not appear to be present, this would be the time to determine if additional resources should be spent on this analysis. Don't spend good money trying to recover from a bad investment!

As there appears to be a strong linear relationship, this would suggest we should attempt to build a linear regression model for our purposes.

Model Building

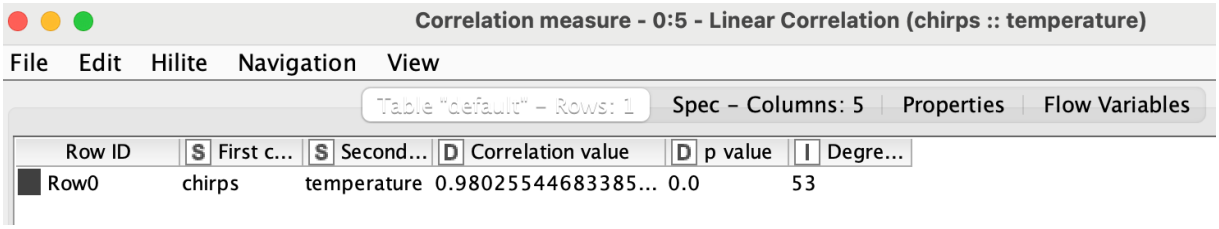
30 pts

To build this model, we'll once again turn to KNIME. The data are imported and ran through a linear regression learner. We then test out the resulting line of best fit to see what we should expect the temperature to be at 40 chirps / 15 seconds. As we are feeding this model known and trusted data, this is an example of supervised learning. In supervised learning, the computer is "taught" what the model should look like and how one feature informs another feature based on a trusted data set.



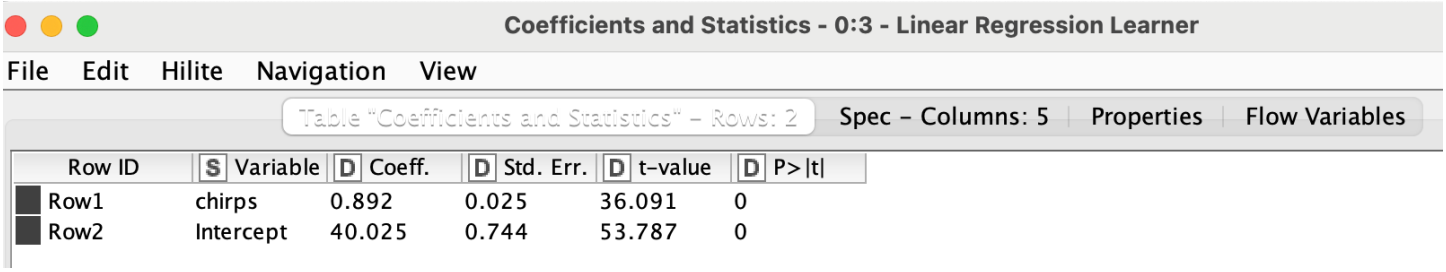
As stated, the end deliverable with this model should be.a Line of Best Fit. This line represents a straight line that, when drawn, yields the smallest squared sum of the distance of all points from the line. (We're squaring these numbers to prevent negative distances cancelling out the positive distances.) It is quite literally the line that best fits all the data.

As expected, KNIME confirms that there is a strong positive correlation between these two sets. A correlation value can go from -1 (perfect negative correlation) to 1 (perfect positive correlation). The further the value drifts from these two ends, the less correlation is found. A value of .98 is a strong indication that we're looking at a linear relationship. (Note, we cannot confirm a causal relationship with this method.)



The line of best fit is thus in the format of Y = mx + b (or Temperature = slope (chirps) + y intercept.

We found the line is y = .892(chirps) + 40.025.



If we were to use this in a practical way, predicting the temperature at 40 chirps / 15 seconds would yield:

Temp = .892(40) + 40.025

Predicted Temp = 75.705 degrees.

Interpretation/Summary
10 pts

A critical step in the data science life cycle is interpreting the findings in relation to our question. In terms of reasonability, this temperature is within the range of what the known data set would call for. It is not outlandishly high, nor is it near 0 kelvin. Determining if the answer is reasonable should be done before progressing!

Since we've determined that the result is reasonable, the next step would be to interpret and communicate the results.

We found that an extremely strong correlation between chirps and temperature. This correlation enabled us to build a linear regression model. This model was then able to give us back an informed estimate as to what the temperature is from a new data point provided. With these observations, we can say that we CAN predict outside temperature based on cricket chirps.