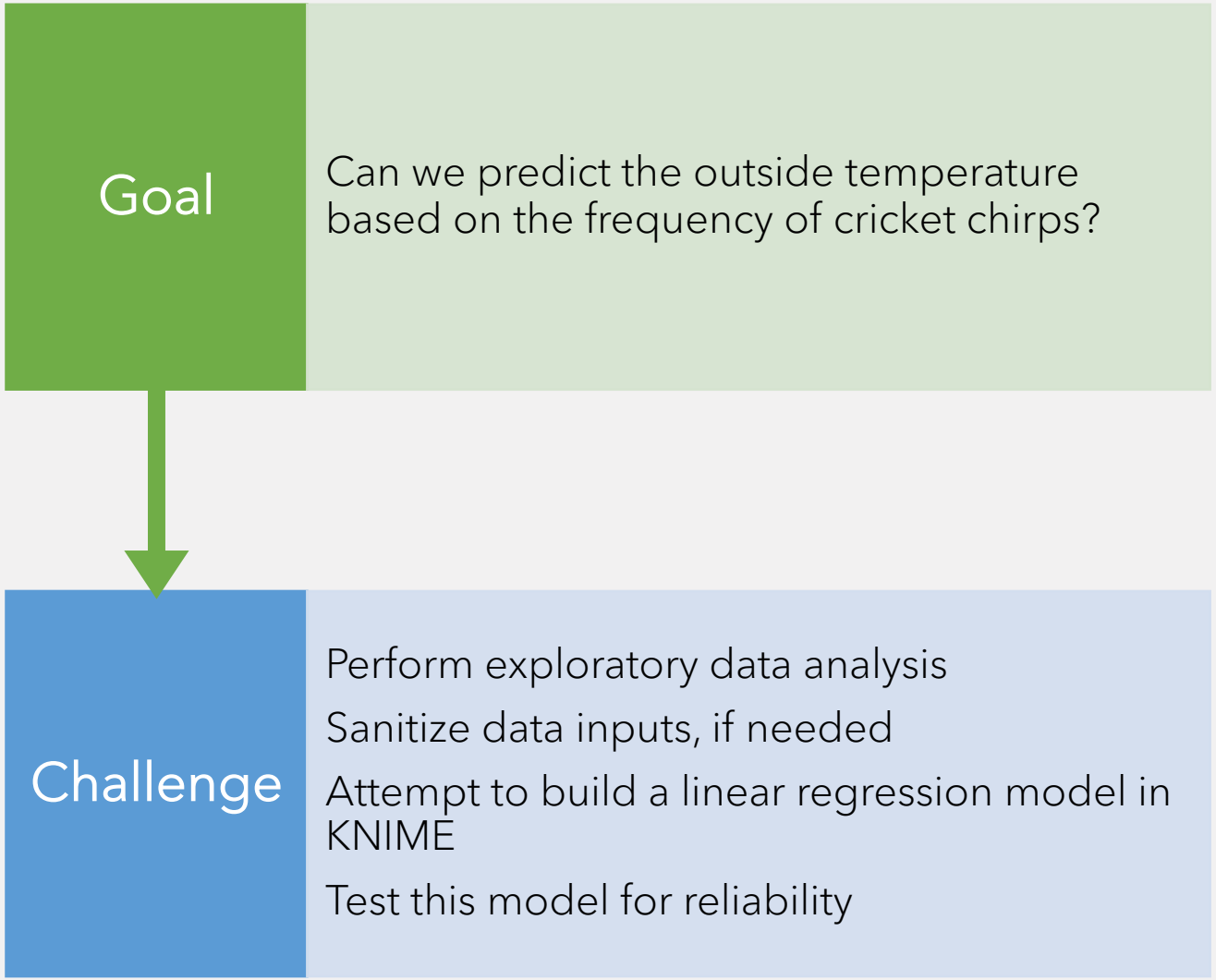
DS 210 : Intro to Data Science Final Project

Robert Daniels, Spring 2022

Introduction and Stating the Question



The Data Set

Initial Findings

- The raw .csv file contains 58 records of type float, with a shape of (59,2) including the attribute labels

  Chirps: cricket chirp frequency (per 15 seconds)

  Temperature: Fahrenheit

- Each attribute contains 1 record with a null value

```
df.isnull().sum()
[223] ✓ 0.1s
...   chirps        1
      temperature   1
      dtype: int64
```

```
df.head()
[222] ✓ 0.1s
...
```

	chirps	temperature
0	44.0	80.5
1	46.4	78.5
2	43.6	78.0
3	35.0	73.5
4	35.0	70.5

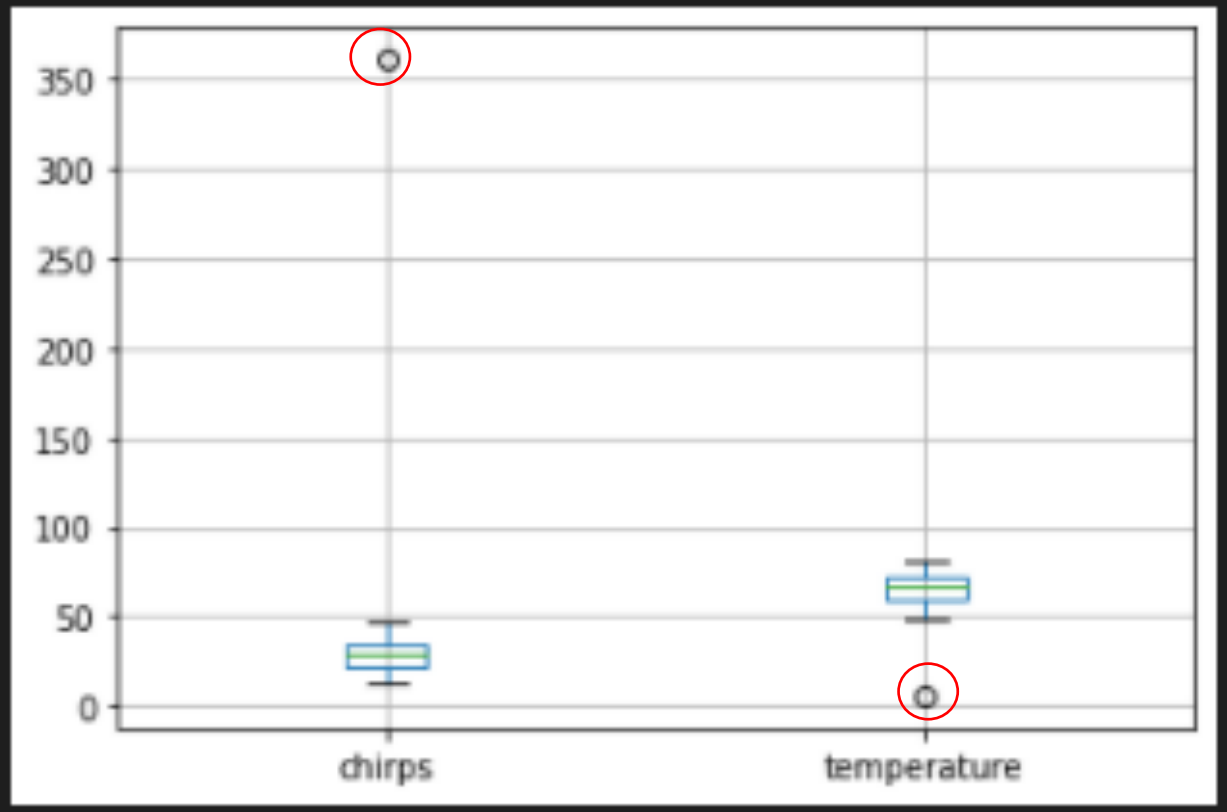
Data Analysis : The messy null

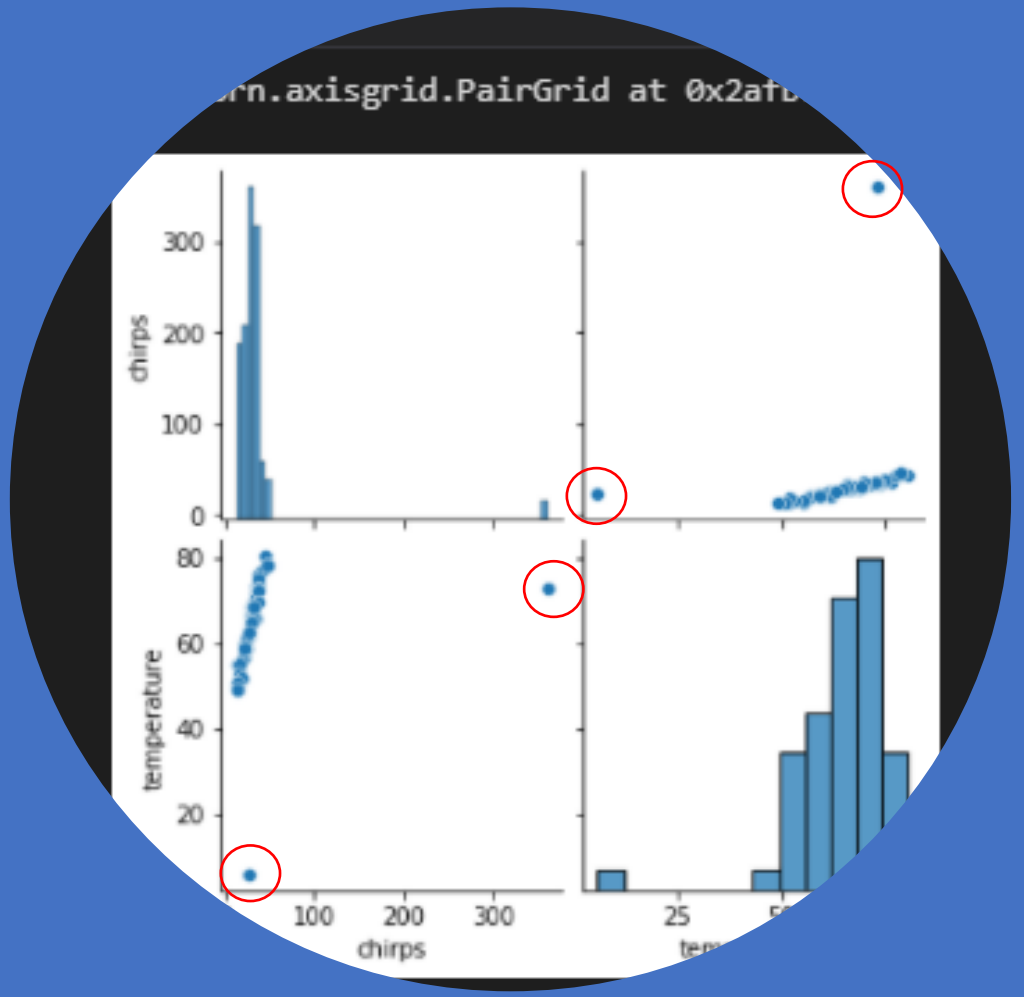
- In this case, the null values represent a small portion of the data set.
- Thus, we can safely remove the 2 records in question.

```
[224]  df = df.dropna()
       ✓ 0.9s

[225]  df.info()
       ✓ 0.1s

...    <class 'pandas.core.frame.DataFrame'>
       Int64Index: 57 entries, 0 to 58
       Data columns (total 2 columns):
        #   Column       Non-Null Count  Dtype
       ---  ------       --------------  -----
        0   chirps       57 non-null     float64
        1   temperature  57 non-null     float64
       dtypes: float64(2)
       memory usage: 1.3 KB
```

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

[227] ✓ 0.3s





Data Analysis: Outliers

Outliers are visually evident in the data set. That won't do!

Data Analysis : Outliers

- Outliers are clearly skewing the data set
- Solution: apply statistical reasoning to find the outliers in question

```
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

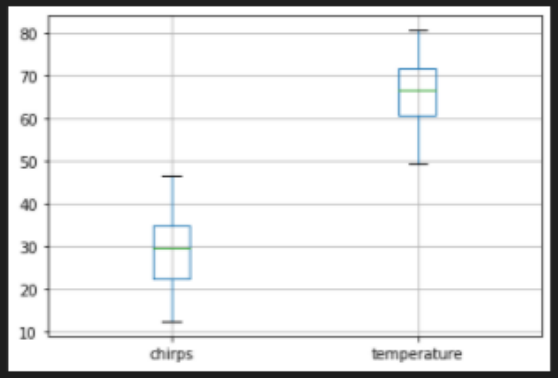
```
This is the bottom:  3.75
This is the top:  53.75
Q3:  35.0
Q1:  22.5
IQR:  12.5
The outliers, if any, were:  [361.0]
```

Data Analysis: Outliers

- Mask out the outliers from the data set
- Much better!
- A linear relationship emerges from the data

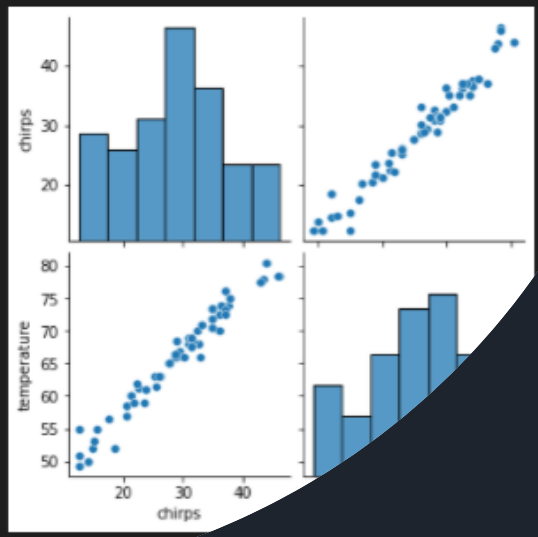
```
df = df[df.chirps < 361]
df = df[df.temperature > 6]
```
[234] ✓ 0.2s

```
boxplot = df.boxplot()
```
[235] ✓ 0.3s




```
sns.pairplot(df)
```
[236] ✓ 1.1s

```
<seaborn.axisgrid.PairGrid at 0x2afb7643a90>
```



Data Analysis: General Statistics

```
df.describe()
```

[253] ✓ 0.2s

	chirps	temperature
count	55.000000	55.000000
mean	28.805182	65.713636
std	8.865180	8.065188
min	12.500000	49.250000
25%	22.375000	60.500000
50%	29.500000	66.500000
75%	35.000000	71.500000
max	46.400000	80.500000

Refining the question: Any reason to change our goal?

Recall

- Our goal was to see if there was a strong correlation between chirp frequency and temperature

Visualize

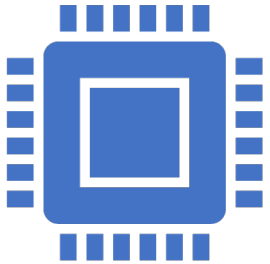
- Visually, a strong linear trend can be seen between these two attributes

Build

- Next step: Use KNIME to build a model

Model Building: KNIME

Using the KNIME platform, we can build a linear regression model based on the data set.

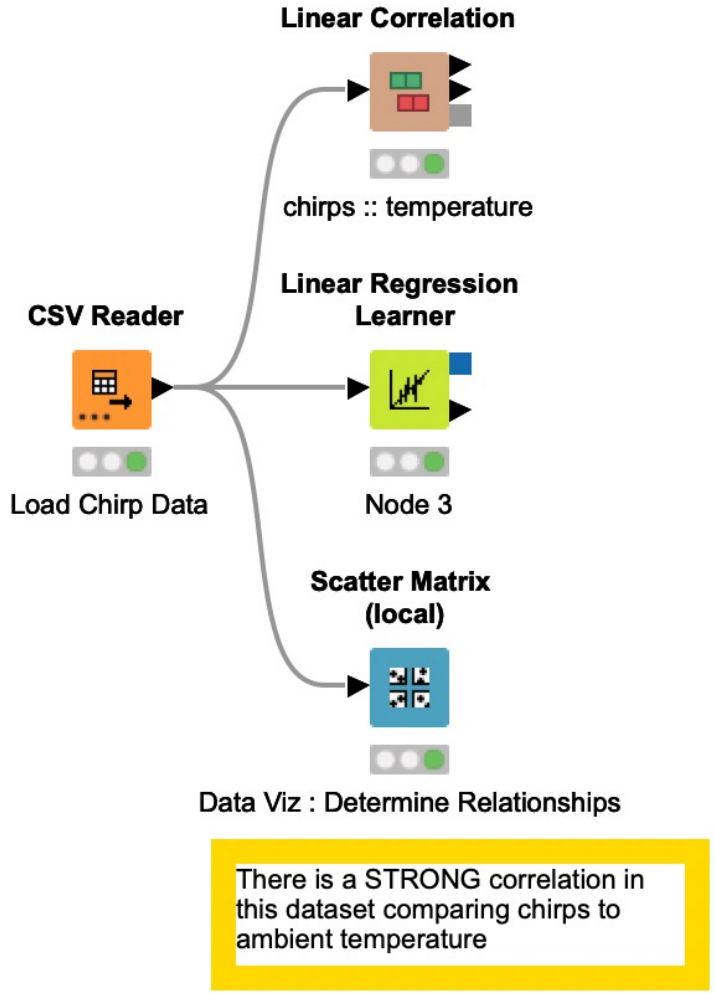


Linear Regression is a supervised learning method.

We provide the computer with a subset of known data
Let the computer build a model off said data
Verify the model with another subset of known data

KNIME: Workflow

- The data are loaded via the CSV reader.
- From this point, a scatter matrix can be generated for visualization
- Basic linear correlation and a linear regression learner can be run on the frame



KNIME: Correlation coefficient and Statistics

- Post processing, the correlation is STRONG, with a correlation value of .98
- This value ranges from 0 to 1, roughly meaning that 98% of the variance in the dependent variable can be explained by a variance in the independent variable

Correlation measure - 0:5 - Linear Correlation (chirps :: temperature)

File Edit Hilite Navigation View

Table "default" – Rows: 1 | Spec – Columns: 5 | Properties | Flow Variables

Row ID	S First c...	S Second...	D Correlation value	D p value	I Degre...
Row0	chirps	temperature	0.98025544683385...	0.0	53

KNIME: Correlation coefficient and Statistics

To find temperature as a function of chirps (temp(chirps)):

temperature(y) = .892(chirps) + 40.025

Coefficients and Statistics - 0:3 - Linear Regression Learner

File Edit Hilite Navigation View

Table "Coefficients and Statistics" – Rows: 2 | Spec – Columns: 5 | Properties | Flow Variables

Row ID	S Variable	D Coeff.	D Std. Err.	D t-value	D P>|t|
Row1	chirps	0.892	0.025	36.091	0
Row2	Intercept	40.025	0.744	53.787	0

KNIME: Regression Line

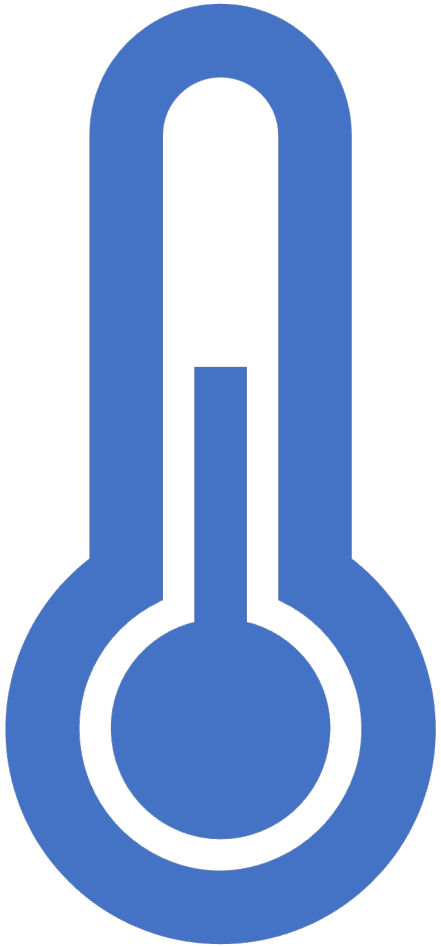
- What is it: the line of best fit
- You are trying to find a line that, when drawn, has the least amount of sum squared distance between each point to the line
- Why squared? Because some points are expected to be under, and some over.
  - Squaring each value eliminates the cancel out
- This would be time consuming to do by hand!
- Thankfully, computers are made for this type of thing
- Y = mx + b
- X (slope)
- b(y intercept)

KNIME: Prediction: 40 chirps

temperature(y) = .892(chirps) + 40.025

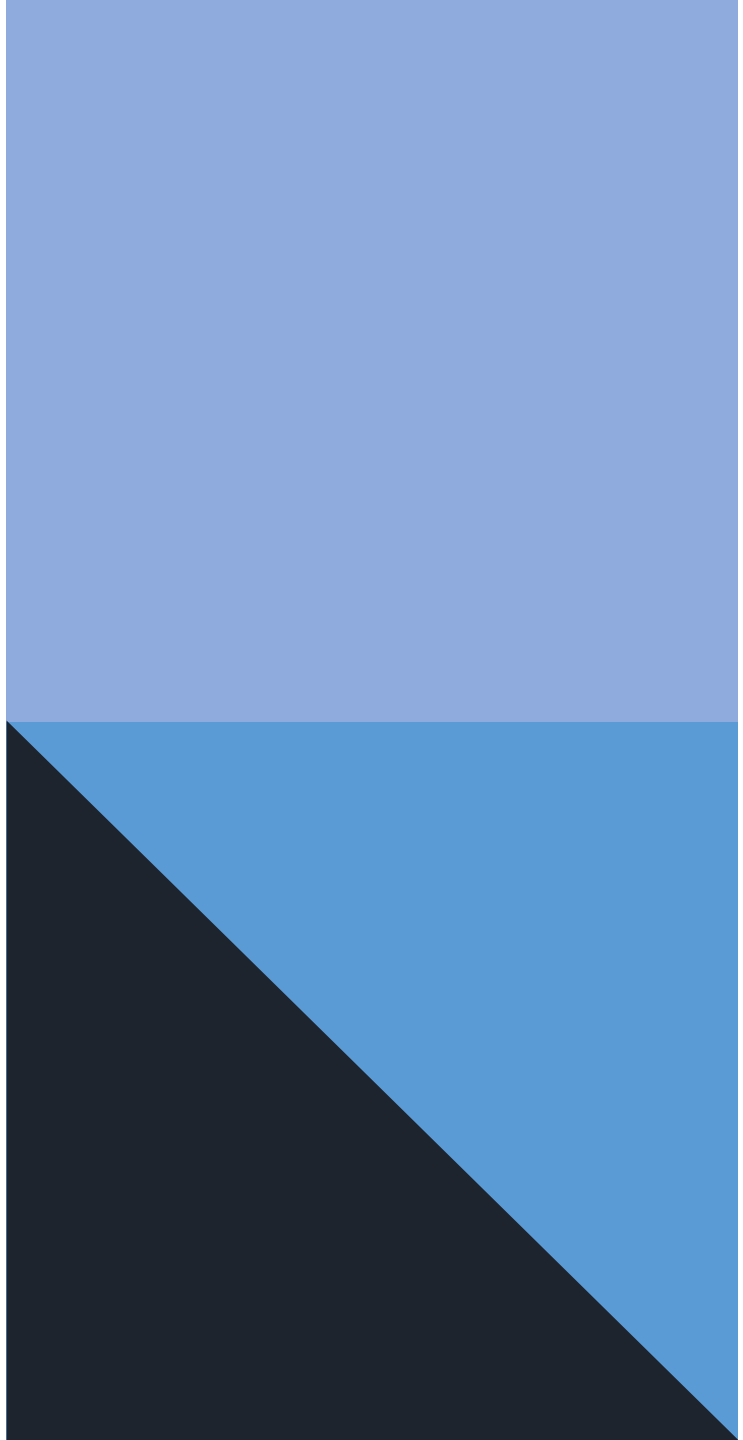
Temperature = .892(40) + 40.025

Predicted Temperature = 75.705 degrees



Reasonableness check! Important!

- The model is predicting at 40 chirps, we'd expect to see a temperature around 75.7 degrees. Given our known data points, this is in line with expectations

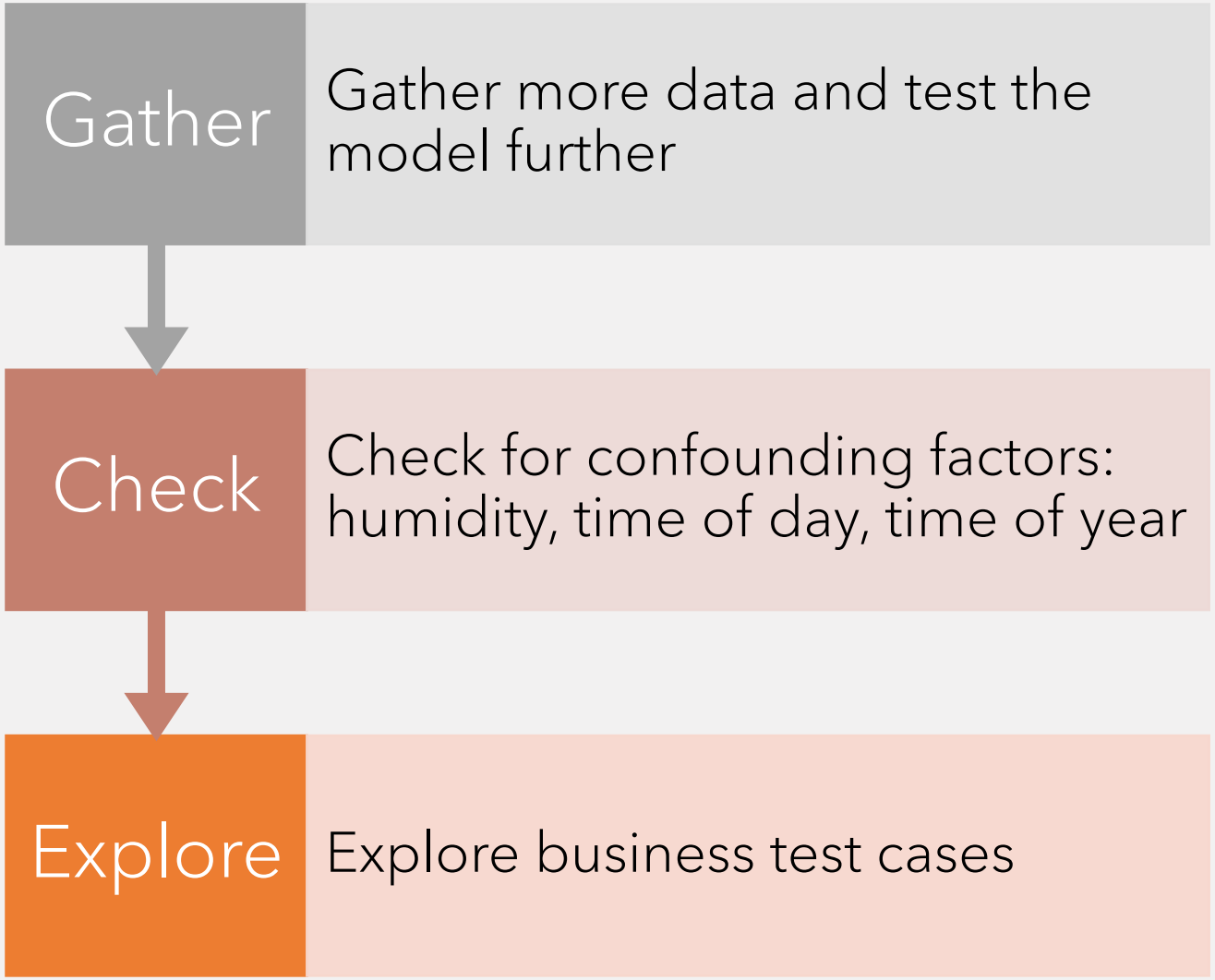


chirps	temperature
36.2	72.5
36.2	70
36.5	74
37	76.25
37	73.25
37.1	72.5
37.5	74
37.8	75
43	77.5
43.6	78
44	80.5
46	78.5

Interpretation

- Given what we've covered, we have a reasonable basis to conclude we CAN predict outside temperature based on chirp frequency

Next Steps


Gather
Gather more data and test the model further
Check
Check for confounding factors: humidity, time of day, time of year
Explore
Explore business test cases

Recap:

We formed our initial question

- Can we estimate outside temperature based on chirp frequency?

Explored our data set

- Determined null values
- Determined outliers present

Scrubbed our product for processing

- Removed null values
- Discarded outliers

Fed our model the data set in KNIME

Validated our data set

- Using the regression formula generated by the model

Interpreted and came to a conclusion based off the model results

- Based off these data, ambient temperature CAN be estimated based off chirp frequency