Data Analysis in Python

NSC 1002

2021.01.22

Michael Kelleher and Beth Wingate

Outline

Code Design

- **Code Design**: Python's original developer Guido van Rossum insightfully says: "code is read much more often than it is written"
- This means how you **write** the code and **how** it works matters just as much as **if it works**
- Examples we give are just one way: usually the way we think you'll understand or the best-practice method
- Simple is usually better, readable above all else

Code Design

- Right now, it's okay if you don't write great code, you're still learning!
- Later (next week, next month, next year, . . .) you can come back and if your code is well written **including comments!**, it can be easily changed, adapted and/or improved
- Eventually, future-you and future-other-people may rely on your code: be kind to them!

Visualization Design

- Scientists (us!) want to **effectively communicate** ideas about "stuff"
- Data visualization is **extremely important** for this communication
- Best science communication tells a story
  - **Beginning**: What is the question, why is it important?
  - **Middle**: What was done, how was it done, what was the result?
  - **End**: What do the results mean, is the question answered, how does it relate?

Robustness and Resistance

- Classical statistical techniques based on strict assumptions about nature of data
- These were made out of necessity for calculations to be performed by hand
- Still can be useful, though, but assumptions should be verified before use
- Methods should be **robust** and **resistant**
- **Robustness** means a measure is not sensitive to particular assumptions about nature of data
- **Resistance** means a measure is not sensitive to a small number of outliers

Numerical Summary Measures

- These reduce a large set of data to a single (or a few) numbers
- They are a way to describe and compare data (under certain conditions)
- Probably already know a few: mean, median, mode, variance, etc.

Numerical Summary Measures: Location

- **Sample mean**: `numpy.mean()`
- **Median**, or the 50th percentile: `numpy.median()`
- **Trimean** depends on quartiles, resistant and robust:

$$TM = \frac{q_{0.25} + 2\,q_{0.50} + q_{0.75}}{4}$$

  See `numpy.percentile`

- **Trimmed mean** also resistant, trims off extremes of data set:

$$\overline{x}_\alpha = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} x_{(i)}$$

  where α denotes the proportion of data excluded

Numerical Summary Measures: Spread

- **Sample standard deviation**: `numpy.std()`
- **Inter-quartile range** depends on quartiles, resistant and robust:

$$IQR = q_{0.75} - q_{0.25}$$

- **Median absolute deviation** analogous to standard deviation, but for median:
  MAD $= \text{median}|x_i - q_{0.5}|$
  (This may be costly to compute for large data sets, but resistant)
- **Trimmed variance** also resistant, trims off extremes of data set:

$$\overline{s}^2_\alpha = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} \left(x_{(i)} - \overline{x}_\alpha\right)^2$$

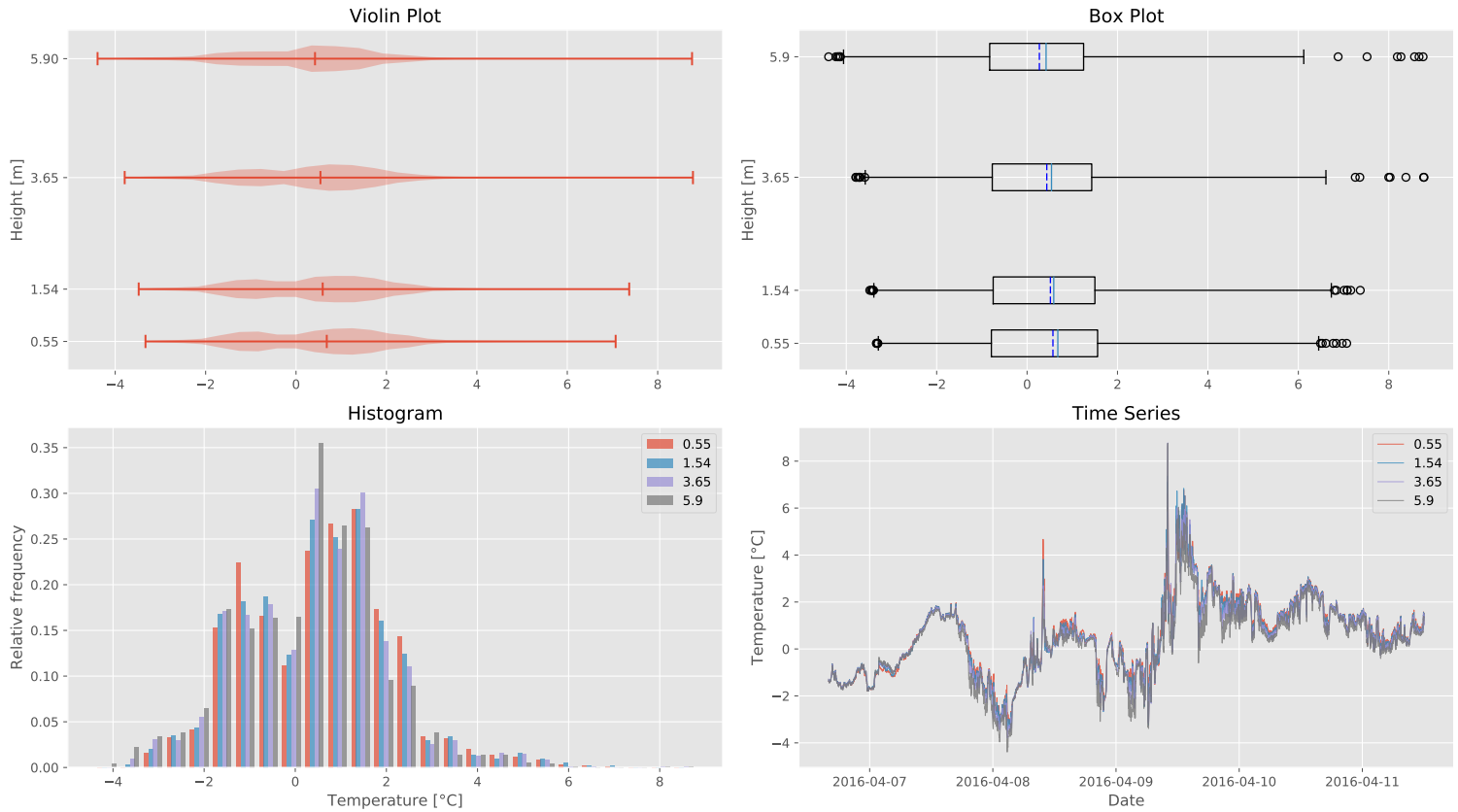
where α denotes the proportion of data excluded

Visualization Design

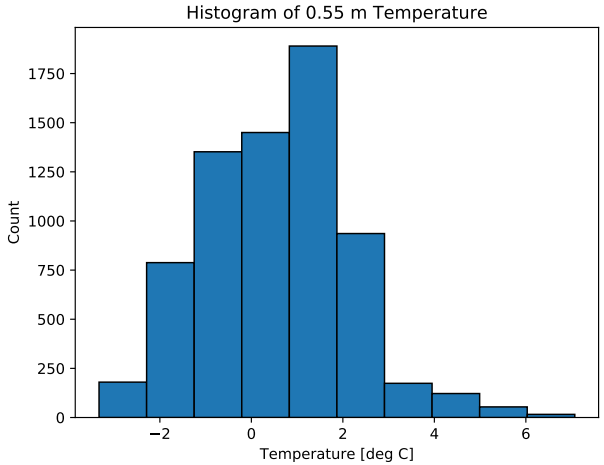
- Data viz comes in the middle, displaying the results in a meaningful sense
- Visualization questions:
  - Who is my audience?
  - What piece of the story does this data tell?
  - What is the best way to communicate that?
  - How will my audience perceive this visualization?

Exploratory plots

Quickest way to see how your data are distributed

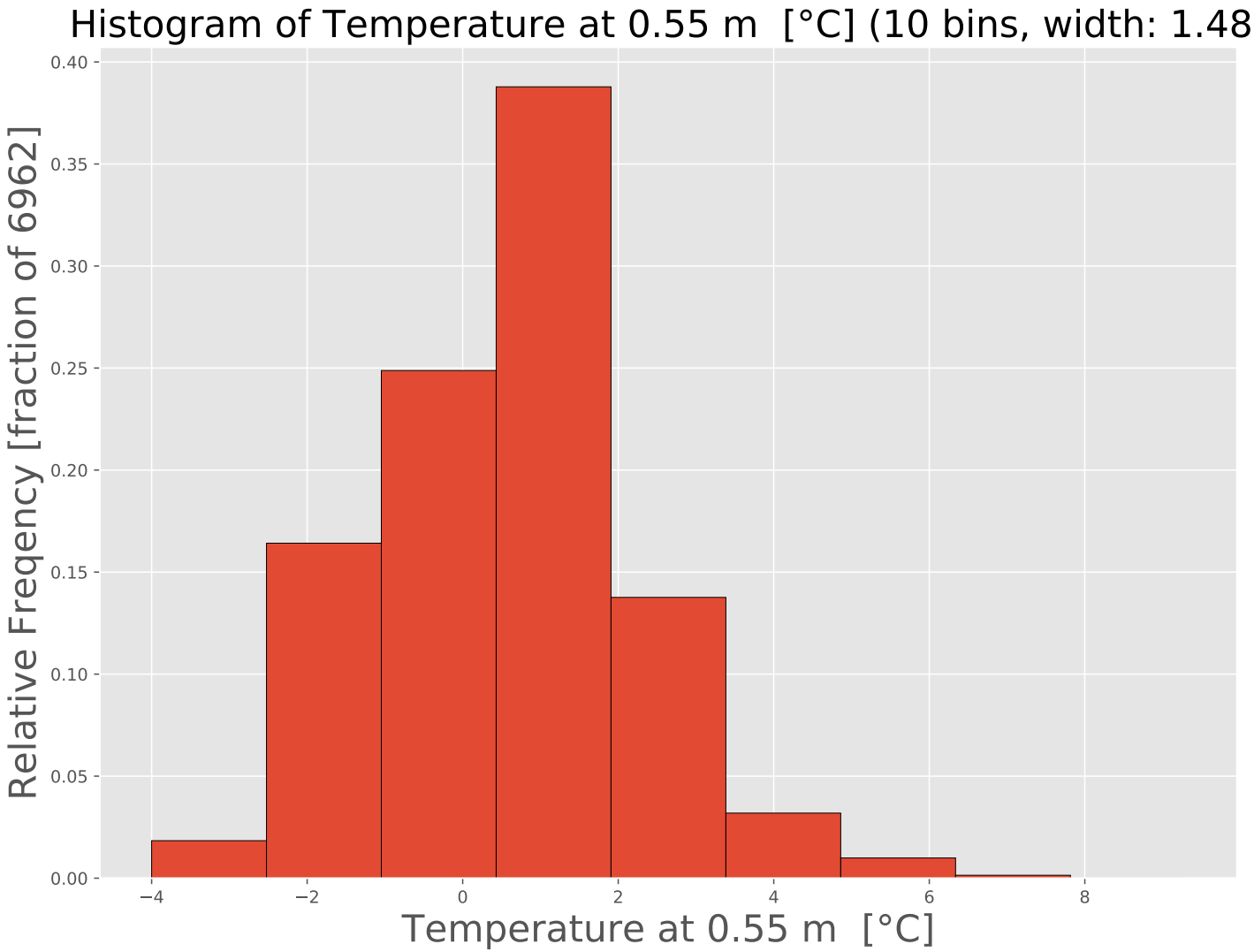


Exploratory Plots: Histogram

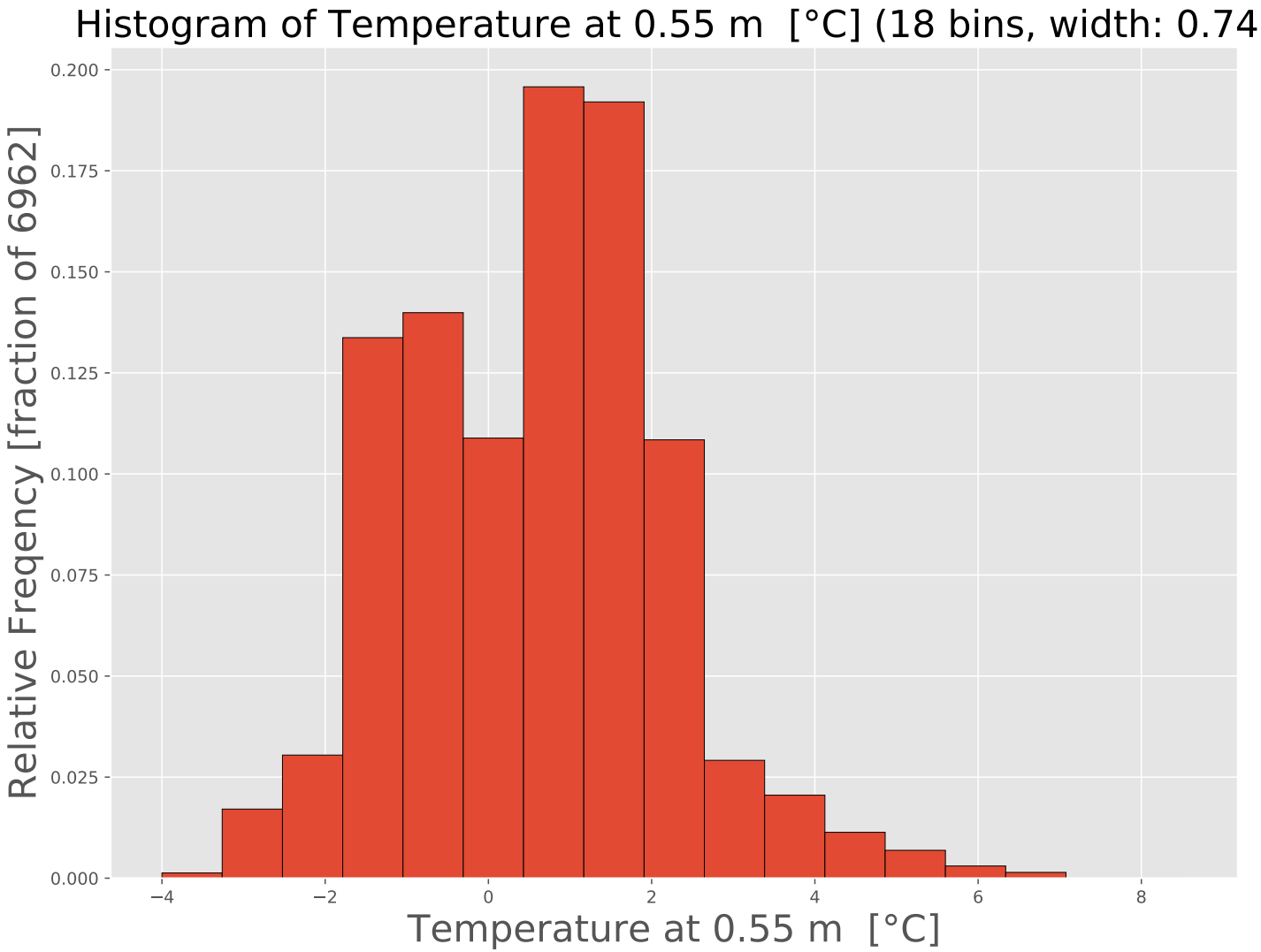



```
import pandas as pd
import matplotlib.pyplot as plt

def main():
    """Load data and plot a histogram."""
    # Load mast data into a DataFrame
    data = pd.read_csv('mast_data.csv')
    # Plot histogram with 10 bins, make box edges black
    plt.hist(data['t0'], 10, edgecolor='k')
    plt.xlabel('Temperature [deg C]')
    plt.ylabel('Count')
    plt.title('Histogram of 0.55 m Temperature')
    plt.show()

# Call main
main()
```

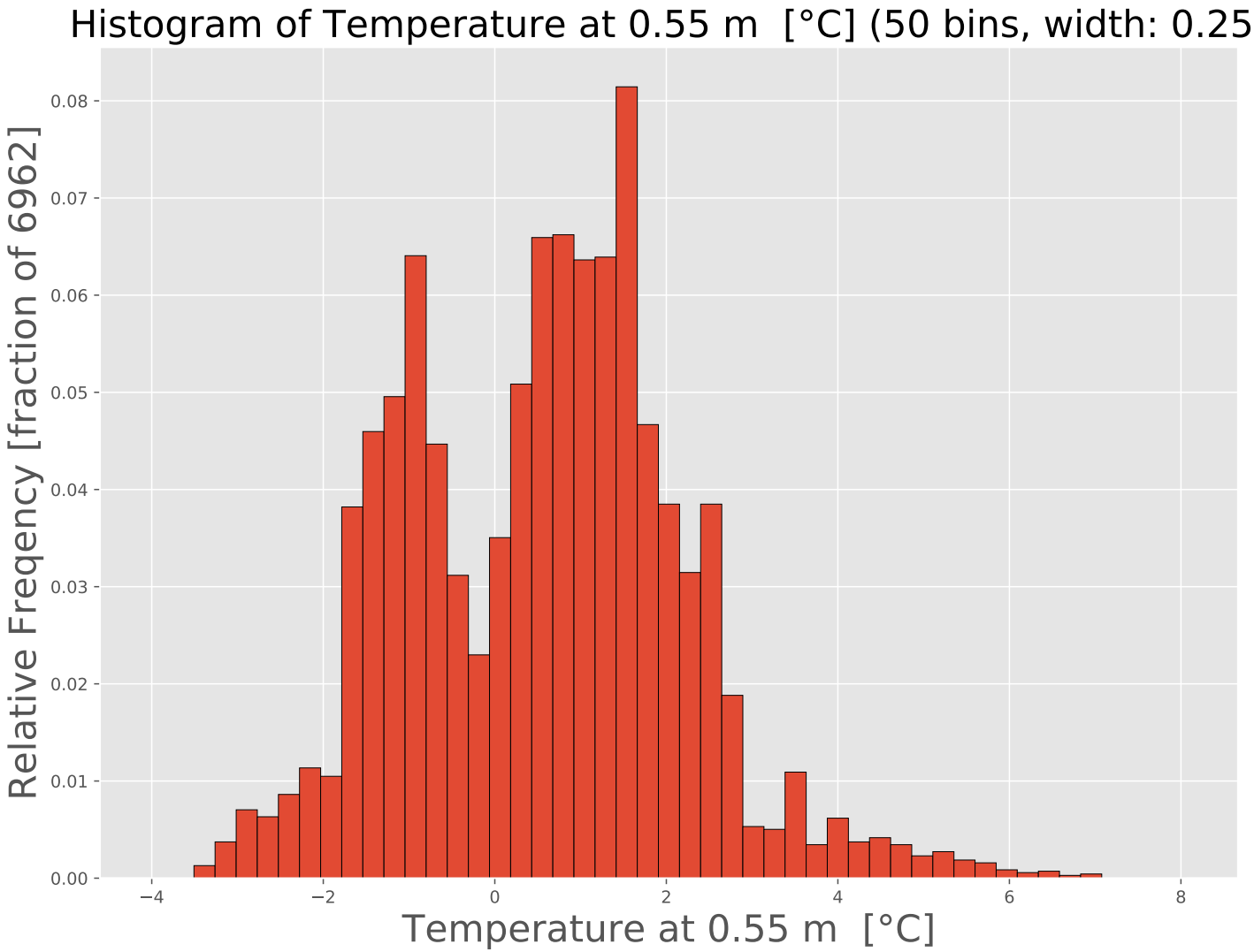
Exploratory Plots: Histogram



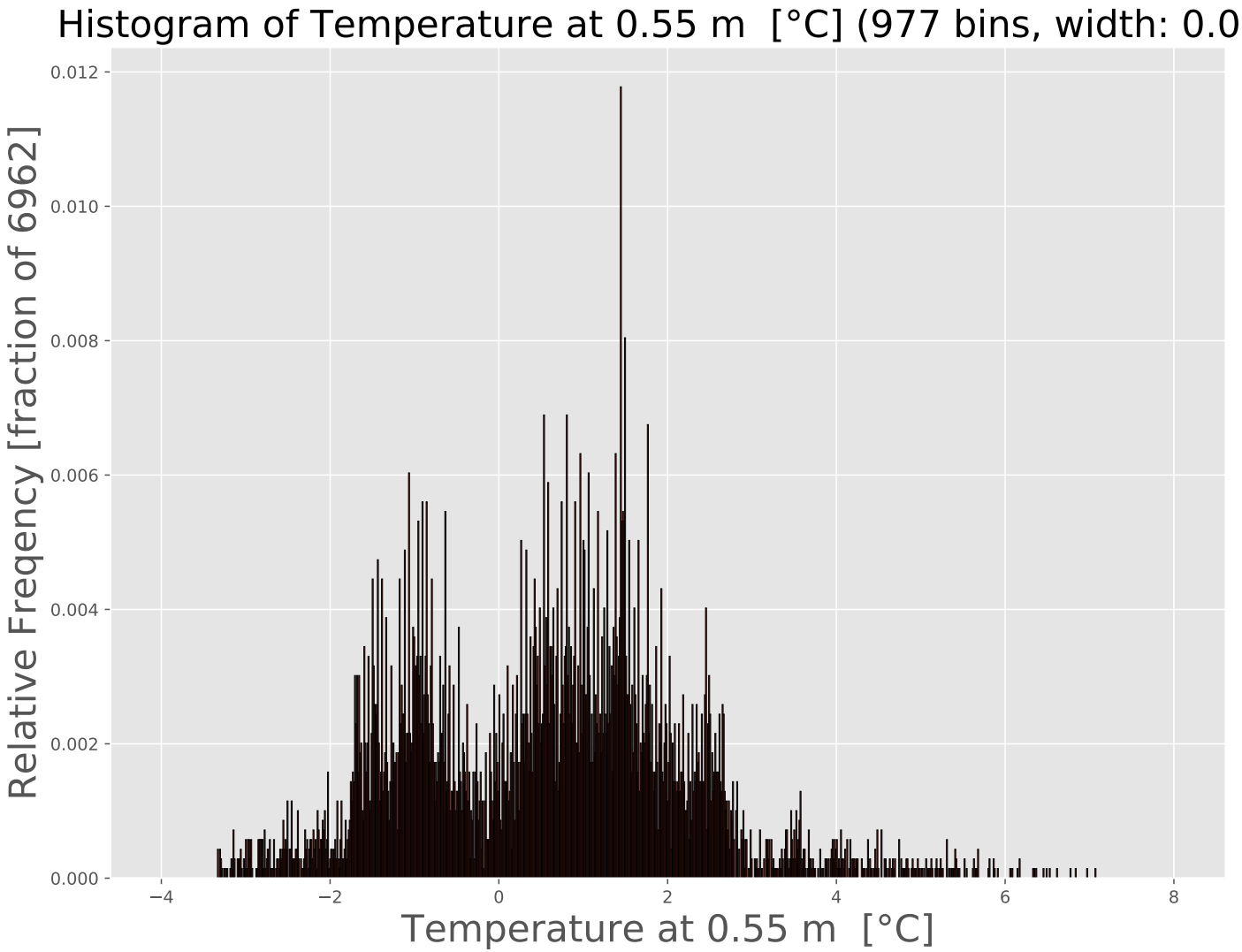
Exploratory Plots: Histogram



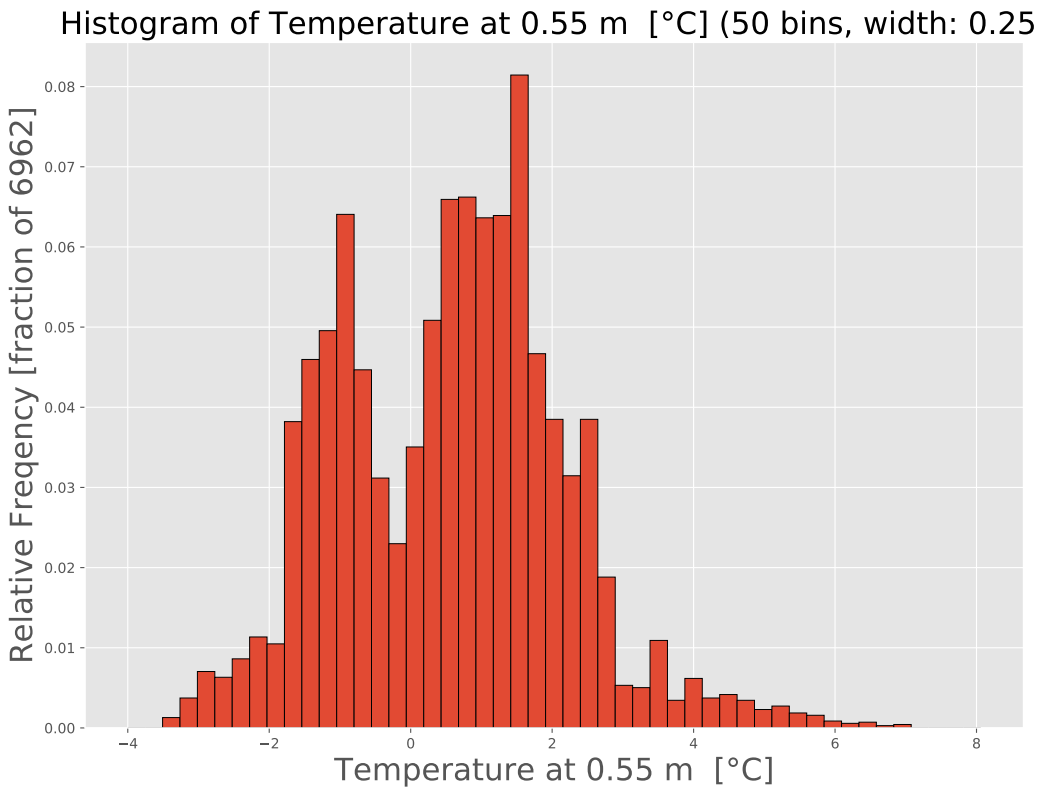
Exploratory Plots: Histogram



Exploratory Plots: Histogram



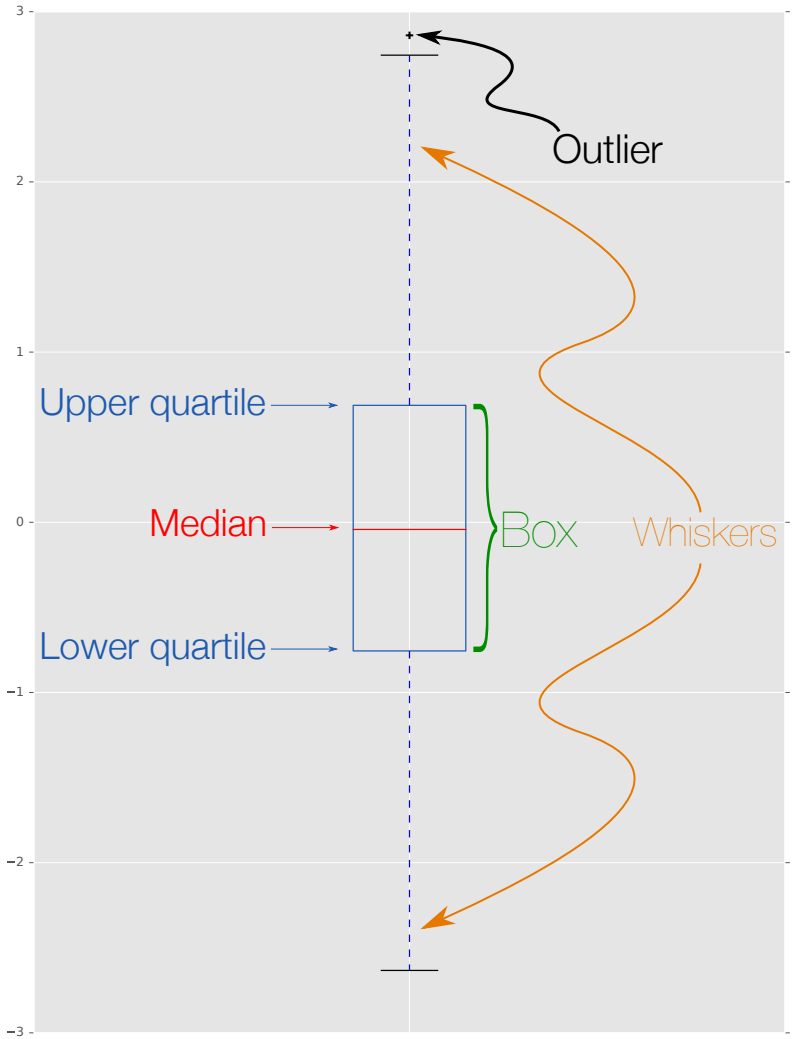
Exploratory Plots: Histogram



$$h \approx \frac{c\,IQR}{n^{1/3}}$$

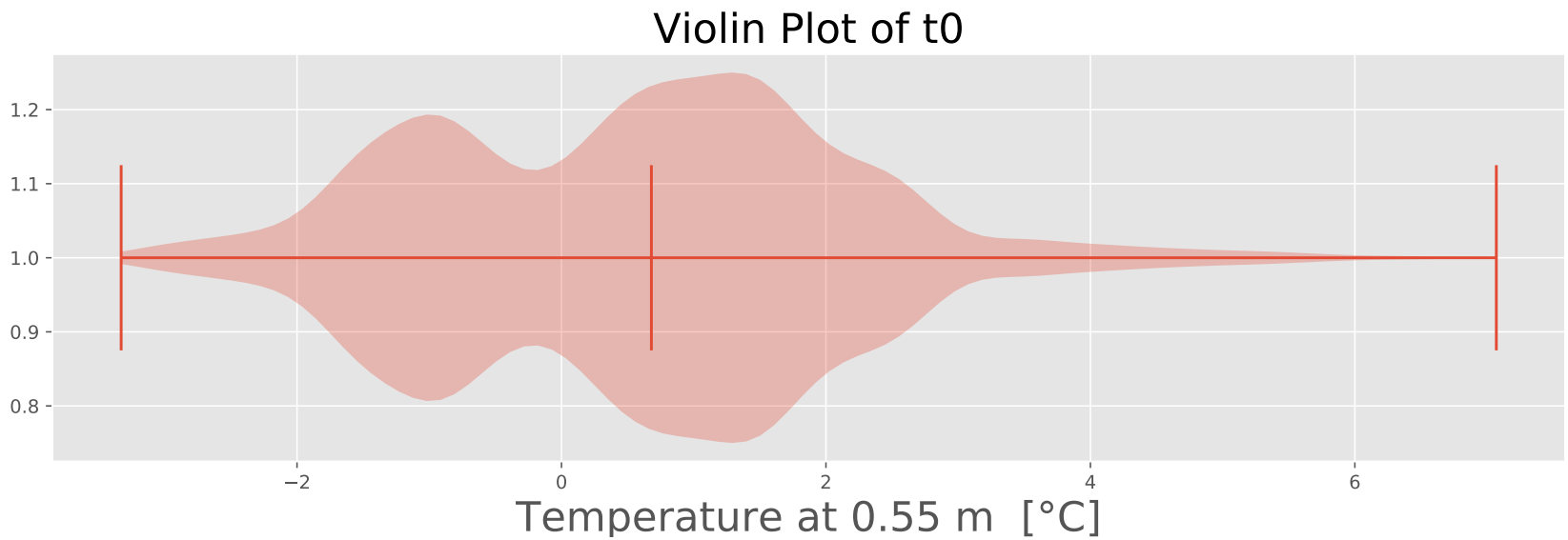
Where $c \in [2.0, 2.6]$, where 2.0 is for skewed/multi-modal data, and 2.6 used for Gaussian data

Box and Whisker Plots




```
import matplotlib.pyplot as plt
import numpy as np
sample_data = np.random.randn(1000)
plt.boxplot(sample_data)
```

Violin Plots




```
import matplotlib.pyplot as plt
import pandas as pd
data = pd.read_csv('mast_data.csv')
plt.violinplot(data['t0'], showmedians=True, vert=False)
```

Wrap Up

- Code and visualisation design (remember future-you!)
- Numerical summaries
  - Location (e.g. mean, median, trimean)
  - Spread (e.g. variance, IQR, MAD)
- Graphical summaries
  - Histograms
  - Box plots
  - Violin plots

References/Resources

- Pandas visualization tools http://pandas.pydata.org/pandas-docs/version/0.20/visualization.html
- Matplotlib Reference http://matplotlib.org/gallery.html
- Design: http://www.informationisbeautiful.net/
- Why Should Engineers and Scientists Be Worried About Color? http://www.research.ibm.com/people/l/lloydt/color/color.HTM