CPE 486/586: Machine Learning for Engineers

04Bonus Data Visualization

Fall 2025

Rahul Bhadani

Outline

Data Exploration

Why Data Exploration?

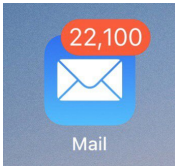
1. Data exploration is useful to answer a number of questions about data, thereby corroborating or invalidating hunches and preconceptions.
2. Data exploration reveals unexpected patterns, trends, and exceptions as well as stimulates new perspectives and insights.

Understanding Different Types of Data and Data Structure

1. What type of data is right for the question you are answering?
    1. In-class exercise: Consider a question you are interested in. What type of data would you be collecting?
2. Extract, use, and organize your data.
3. Make sure the data is relevant and valid.
4. Learn how data is generated & collected.
5. Different formats, types, and structures of data.
6. What does clean data mean?

Ubiquity of Data in the Modern World

Data is generated all around the world in the form of texts, pictures, videos, emails, through social media, mobile phones, etc.



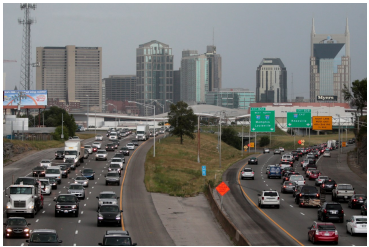
Each image pixels are data, each video frames are data.

Collecting Data through Survey and Experiments

In addition, we can also collect data. An example is the US Census Bureau which collects data for making informed decisions about future policies to help businesses and citizens.

Running surveys is one of the prime sources of data collection in healthcare.

Field experiments to collect data: for example cars with on-board sensors as well as augmented sensors driving on streets provide information about traffic, pedestrians, etc.



How Data will be Collected

Consider an example question: “What is causing increased rush hour traffic in your city?”

The first step we can think of is to validate whether we get traffic congestion during rush hour. We observe the traffic pattern by counting the number of cars on city streets during particular times.

Data may tell us that cars are being backed up on specific streets.

Data Sources I

First party

This is the data you collect through experiments and surveys. Consider this to be the most reliable as you know how you are collecting this data.

Data Sources II

Second party

Second-party data is collected by a group directly from its audience and then sold. For example, if you are unable to collect your data because conducting a new experiment is hard, time-consuming, and logistically challenging, you might be able to buy it from an organization or a group that has already led the traffic pattern study in your city.
The data is still reliable as it comes from a source that has some experience in traffic analysis.
You may get traffic data from a research lab that recently concluded a traffic experiment.
Example: https://i24motion.org/
You can request an account at https://i24motion.org/ to download relevant datasets.

Data Sources III

Third party

This type of data is collected from outside sources who did not collect it directly or they didn't collect it for the purpose you are going to use the data.

- Third-party data comes from a number of different sources.
- Data needs to be investigated further for accuracy, bias, credibility, and trustworthiness before it can be used as it is not as reliable as second or first-party data.
- Example of traffic data: from HERE Map API, Google Map API; and other data aggregator services. Data sold by various companies who have aggregated user data from various social media, by the fine prints of T&C of online contracts, etc.

How much data to collect?

When you're gathering your own data, it's crucial to make sensible choices about the sample size. For some projects, a random selection from existing data may suffice. However, other projects may require a more targeted approach to data collection, concentrating on specific criteria. Remember, each project has unique requirements.

Population

Definition

All possible values in a certain dataset. **Example: If analyzing data about car traffic in a city, your population is all cars in the area.**

How much data to collect?

However, collecting data from the entire population can be challenging, and may not be feasible at all.

Sample

Definition

A part of a population that is a representative of the population.
Example: You might collect data about traffic from one spot in the city and analyze or you might pull random samples from all existing data in the population.

Data Formats

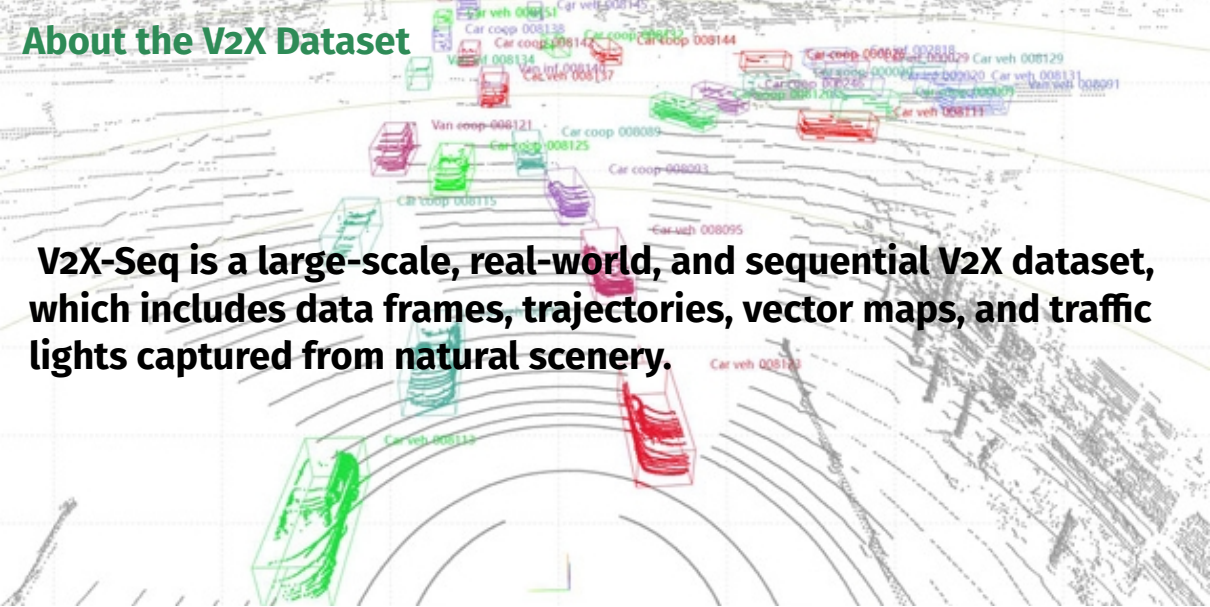
1. **Quantitative Data:** Can be measured, counted, and expressed as numbers.
    1. Discrete
    2. Continuous
2. **Qualitative Data:** Name, category, description – cannot be measured or counted
    1. Nominal: can be categorized without a set order. Example: {yes, no, not sure}
    2. Ordinal: set order is important. Example: movie ratings 1-5 stars.

Data Format Hands On

We will use some example datasets from **A Large-Scale Sequential Dataset for Vehicle-Infrastructure (V2X) Cooperative Perception and Forecasting**: https://github.com/AIR-THU/DAIR-V2X-Seq/.

Download my local copy of the example dataset from https://drive.google.com/file/d/13jvC5YhMpdqlXyrK-IJJuEoZBmAlNzOz/view?usp=sharing.

About the V2X Dataset



V2X-Seq is a large-scale, real-world, and sequential V2X dataset, which includes data frames, trajectories, vector maps, and traffic lights captured from natural scenery.

Reading the Data from Python I

We will be reading the Trajectory forecasting dataset with an infrastructure view using the Panda package from Python.

```
import pandas as pd
datadir = '../cooperative-trajectories/train/data/'
datafile = '1002.csv'

df = pd.read_csv(datadir + datafile)
df.head() # to display a portion of data
```

Reading the Data from Python II

	city	timestamp	id	type	sub_type	tag	x	y	z	length	width	height	theta
0	PEK	1.644129e+09	23	VEHICLE	CAR	AGENT_5	417658.847934	4.730580e+06	0.0	4.098116	2.103868	1.530403	0.578752
1	PEK	1.644129e+09	23	VEHICLE	CAR	AGENT_5	417658.845406	4.730580e+06	0.0	4.098116	2.103868	1.530403	0.578730
2	PEK	1.644129e+09	23	VEHICLE	CAR	AGENT_5	417658.845556	4.730580e+06	0.0	4.098116	2.103868	1.530403	0.578607
3	PEK	1.644129e+09	23	VEHICLE	CAR	AGENT_5	417658.846020	4.730580e+06	0.0	4.098116	2.103868	1.530403	0.578402
4	PEK	1.644129e+09	23	VEHICLE	CAR	AGENT_5	417658.845729	4.730580e+06	0.0	4.098116	2.103868	1.530403	0.578044

Which of the above columns are quantitative data, and which of them are qualitative data?

Structured and Unstructured Data

1. **Structured Data:** Data organized in a certain format such as rows and columns.
   - Spreadsheets
   - Relational databases
   - JSON format datasets

   Structured data works nicely within a data model, which is a model that is used for organizing data elements and how they are related to one another.

   Data elements

   A piece of information such as student's name, A# number, residential address, etc.

2. **Unstructured Data:**
   - Audio
   - Video
   - Emails
   - Photos

Data Model

- The Data model helps to keep data consistent and provides a map of how data is organized.
- Having a data model makes it easier for data engineers and analysts to communicate about data to other stakeholders to make sense of their data and use it to make business decisions.
- Helpful in creating visualization tool.

An example of a data model can be seen in: https://github.com/AIR-THU/DAIR-V2X-Seq/blob/main/dataset/v2x-seq-tfd/README.md.

Metadata

Definition

An abstract concept used to describe your data. **Example:** Smartphone picture taken from your phone has metadata such as image size, ISO, geolocation tagged, timestamp of the image acquisition, etc.

Data Visualization

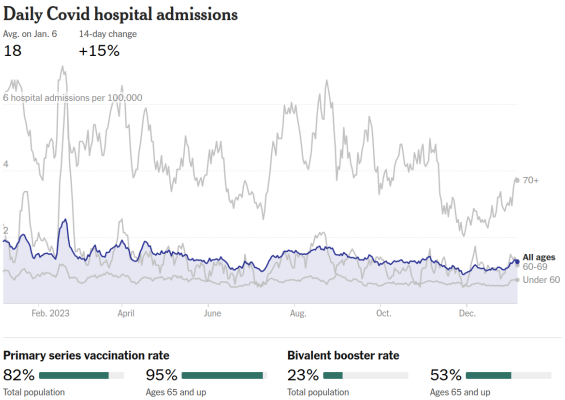
Why Data Visualization?

Being able to present data and insights in a clear, engaging, and accessible manner is a key competency for successful data scientists and analysts. This skill transforms data into a captivating narrative.

Why Data Visualization?

1. Highlights and reveals patterns, trends, and relationships.
2. Simplifies the data for stakeholders.
3. Presents data for better understanding and interpretation

Power of Data Visualization



Source: https://www.nytimes.com/interactive/2023/us/hawaii-covid-cases.html

Data Visualization Best Practices

1. Choosing the right type of visualization.
   - Different types of data require different visualization techniques
2. Keep it simple
   - Line chart for trends and bar chart for comparison
3. Label axes properly, use clear formatting
   - Use a suitable title and legend
4. Less is more.

Example

https://miro.medium.com/v2/resize:fit:640/format:webp/1*ZF-3-ih4QwSVTVXZeVV-iA.gif

Broader Types of Data Visualization

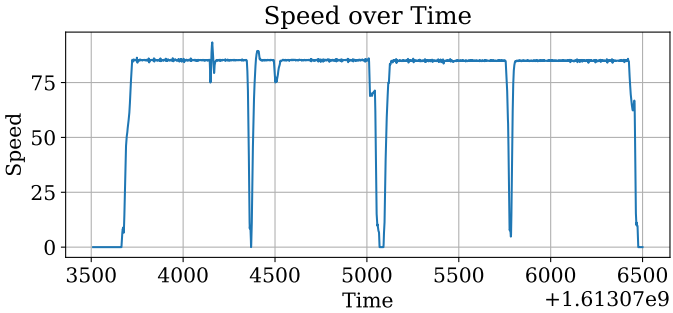
1. Graphical representation of data and information
2. Basic charts and graphs
3. Interactive dashboards and maps

Most Basic Types of Plots

1. Line Plot
2. Bar Plot
3. Scatter Plot
4. Histogram
5. KDE Plot
6. Box Plot

Line Plot I

1. Displays trend over time
2. Connects two subsequent points in the dataset using a straight line
3. The visual of the line plot depends on the order of data points.
4. Illustrate cause-effect relationship.



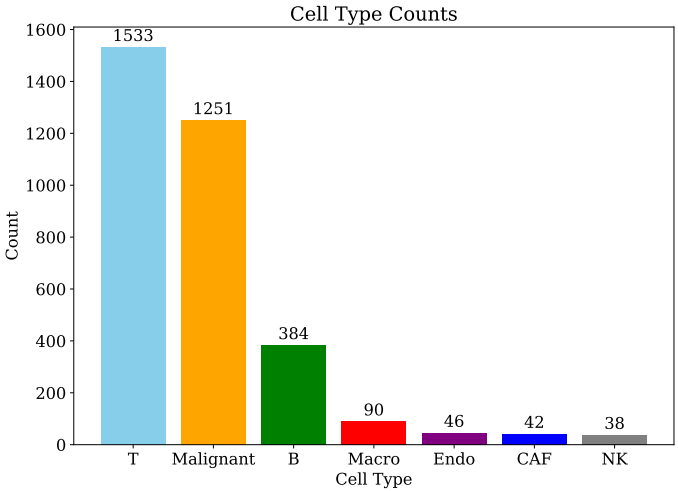
Download a Toyota RAV4 driving dataset from https://drive.google.com/file/d/1UPRgg9HFNlcvud31kTfOI7Zc59x0J8Z_/view?usp=sharing

Line Plot II

```
data_df = pd.read_csv(datadir + datafile, index_col=0)
data_df.head()
# Create the plot
plt.figure(figsize=(8, 3))  # Increase the size of the plot
plt.plot(data_df['Time'], data_df['speed'])
# Add labels and title
plt.xlabel('Time')
plt.ylabel('Speed')
plt.title('Speed over Time')
# Show grid
plt.grid(True)
```

Bar Plot I

1. Displays data using rectangular bars
2. Height of data represents the magnitude of data
3. Helpful in displaying data that has different categories
4. Compare different categories and groups
5. Can also visualize data that can be ranked or ordered



Bar Plot II

Download a single-cell sequencing melanoma dataset from https://drive.google.com/drive/folders/1XepEHBAkpdi7_uUy3cUC12G1ckACw3Dv?usp=sharing. They contain single-cell samples, with genes as features and each entries are gene-expression levels.

Original data source:
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE72056.

Reference:

Introduction to Single-Cell Analysis with Bioconductor,
https://bioconductor.org/books/3.14/OSCA.intro/getting-scrna-seq-datasets.html

Bar Plot III

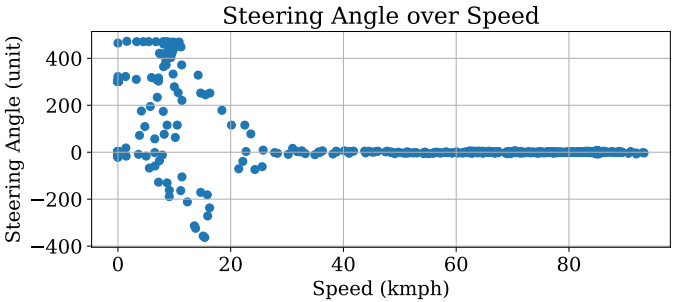
We will visualize how many of each of the cell types are present using a bar plot.

Bar Plot IV

```
counts = data_df['type'].value_counts()
plt.figure(figsize=(10,7))
# Set the global font to be Serif, size 15
plt.rcParams['font.family'] = 'Serif'
plt.rcParams['font.size'] = 15
# Define a color list
colors = ['skyblue', 'orange', 'green', 'red', 'purple', 'blue', 'gray']
bars = plt.bar(counts.index, counts.values, color=colors)
# Adding counts on the top of each bar
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x()+bar.get_width()/2, yval+10, yval, ha='center', va='bottom')
plt.xlabel('Cell Type')
plt.ylabel('Count')
plt.title('Cell Type Counts')
```

Scatter Plot I

1. Good to visualize one variable against another variable
2. The ordering of data doesn't affect the visualization
3. Helpful in detecting outliers and unusual observations
4. Compare different categories and groups
5. Identify clusters or groups in the data



We will visualize how the steering angle is related to speed in the case of the RAV4 driving dataset.

Scatter Plot II

```
plt.scatter(data_df['speed'], data_df['steer_angle'])

# Add labels and title
plt.xlabel('Speed')
plt.ylabel('Steering Angle')
plt.title('Steering Angle over Speed')
```

Histogram I

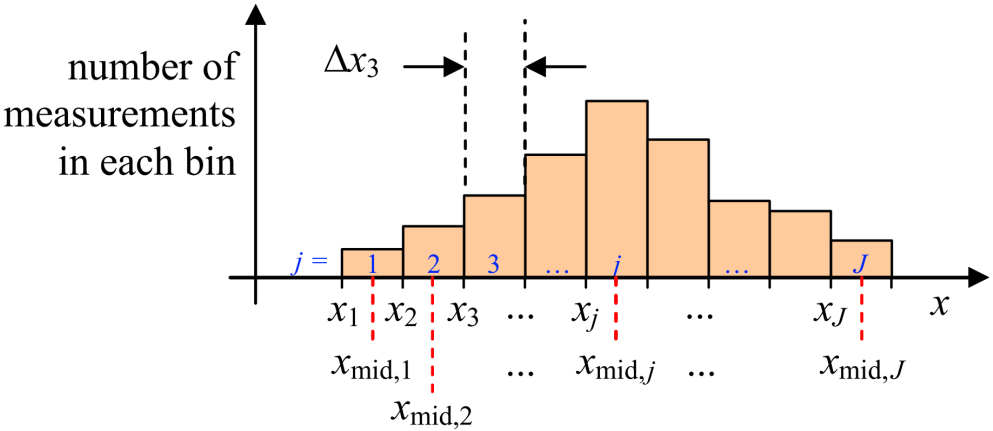
A special type of bar graph that displays variations in data like time, speed, weight, etc. A histogram enables a team to recognize and analyze patterns in data that are not apparent simply by looking at a table of data, or by finding the average or median.

Histograms are approximations of data distribution.

Histograms are constructed by dividing n measurements of a sample into b bins or classes (also called intervals). In such a case, we have the first bin $b = 1$ representing a data range for $x_1 < x < x_2$, the second bin $b = 2$, $x_2 < x < x_3$, and so on.

The middle point of each bin j is $x_{mid,j} = \frac{x_j + x_{j+1}}{2}$. The number of observations within each bin j, called frequency is plotted as a bar graph for each bin.

Histogram II



Histogram III

How to choose the number of bins?

1. Sturgis rule: $B = 1 + 3.3 \log_{10} n$
2. Rice rule: $B = 2n^{\frac{1}{3}}$

Probability Histogram

If you divide the vertical axis (the frequency axis) by the total number of measurements, the resulting histogram is called **probability histogram**. We can also define

$$\text{probability}_b = \frac{\text{the number of measurements in the bin j}}{n} \tag{1}$$

Histogram IV

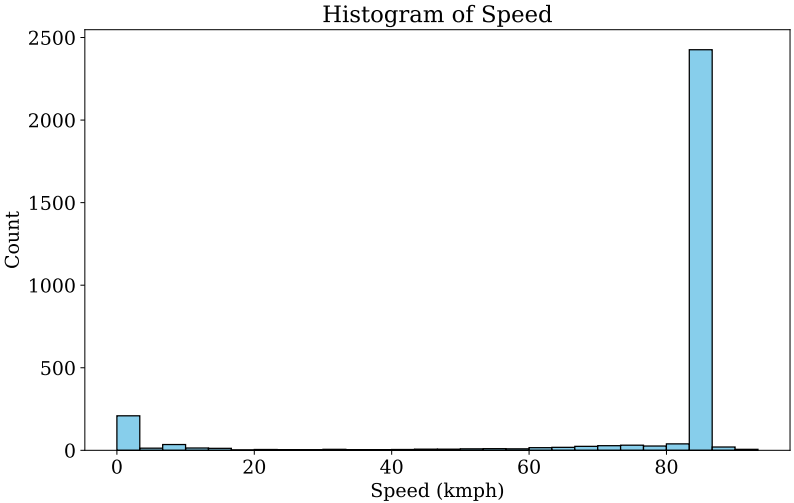
Vertically Normalized Histogram

A histogram obtained by further dividing the vertical axis by the bin width is called a vertically normalized histogram. interval width. The vertical axis of the the vertically normalized histogram is defined as

$$f(x_{mid,j}) = \frac{\text{the number of measurements in the bin j}}{\Delta x_j \cdot n} \tag{2}$$

This makes sure that mathematically the area of the bin is equal to the probability that x lies in that bin.

Histogram V


Histogram of Speed
Count
2500
2000
1500
1000
500
0
0
20
40
60
80
Speed (kmph)

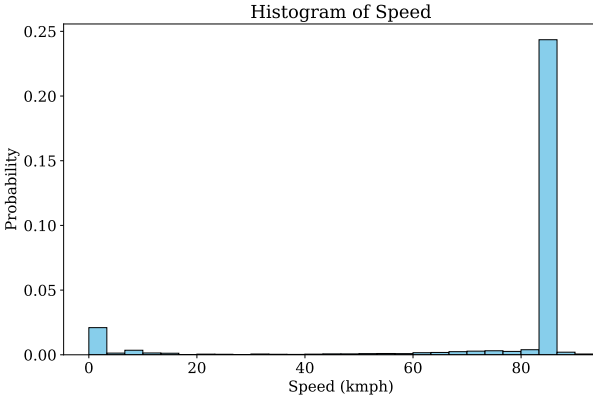
Histogram Plot

```
n = len(data)
num_bins = int(2 * n**(1/3))
plt.figure(figsize=(10,6))
plt.hist(data, bins=num_bins, color='skyblue', edgecolor='black')
plt.xlabel('Speed (kmph)')
plt.ylabel('Count')
plt.title('Histogram of Speed')
```

Vertically Normalized Histogram Plot

Set `density=True` to create vertically normalized histogram

```
n = len(data)
num_bins = int(2 * n**(1/3))
plt.figure(figsize=(10,6))
plt.hist(data, bins=num_bins,
    density=True, color='skyblue',
    edgecolor='black')
plt.xlabel('Speed (kmph)')
plt.ylabel('Probability')
plt.title('Histogram of Speed')
```



Further Reading:

Histogram, https://www.me.psu.edu/cimbala/me345/Lectures/Histograms.pdf

Kernel Density Estimation Plot I

Kernel Density Estimation (KDE)

Kernel density estimation's goal is to estimate the probability density function from a given dataset. The KDE is given by

$$\hat{p}_n(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) \tag{3}$$

where $K(x)$ is called **kernel function** which is generally smooth and symmetric. h is called smoothing bandwidth which controls the amount of smoothing. X_i is a data point. n is the total number of data points. KDE smoothes each data point into a small density bumps and them sum all these bumps together to obtain the final density estimate.

Kernel Density Estimation Plot II

Smoothing Bandwidth in KDE

If we have a smaller bandwidth h, it will result in under smoothing, more wiggly curve. If h is too large, it will cause over-smoothing and some important structures might not be revealed.

Kernel Function

A kernel function should have following properties:

1. $K(x)$ must be symmetric.
2. $\int K(x)dx = 1$.
3. $\lim_{x\to-\infty} K(x) = \lim_{x\to+\infty} K(x) = 0$.

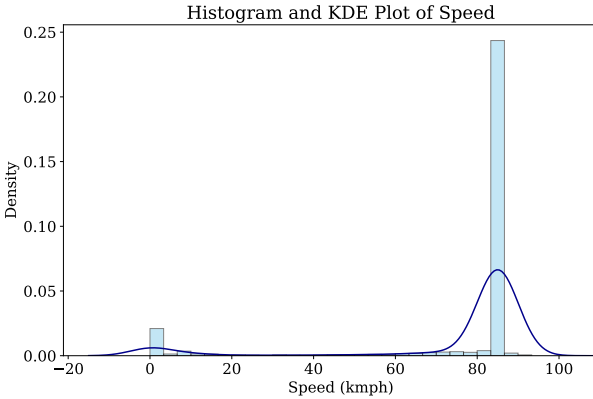
Kernel Density Estimation Plot III

Some Common Kernel Functions

1. Gaussian: $K(x) = \frac{1}{\sqrt{2\pi}e^{-x^2/2}}$.
2. Uniform: $K(x) = \frac{1}{2}I(-1 \leq x \leq 1)$, I is an identity function.
3. Epanechnikov: $K(x) = \frac{3}{4} \cdot \max(1 - x^2, 0)$.

Kernel Density Estimation Plot with Python I

```
n = len(data)
num_bins = int(2 * n**(1/3))
plt.figure(figsize=(10,6))
# Plot histogram
plt.hist(data, bins=num_bins, density=True,
    color='skyblue', edgecolor='black',
    alpha=0.5)
# Overlay KDE plot
sns.kdeplot(data, color='darkblue',
    bw_adjust=1.0)
plt.xlabel('Speed (kmph)')
plt.ylabel('Density')
plt.title('Histogram/KDE Plot of Speed')
```



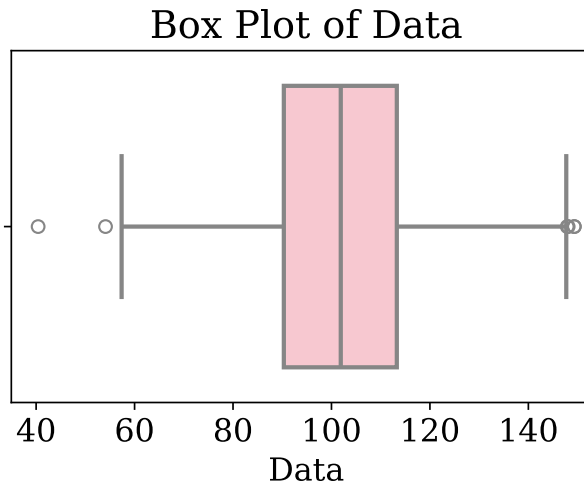
Kernel Density Estimation Plot with Python II

Further Reading:

Fast & Accurate Gaussian Kernel Density Estimation,
https://web.archive.org/web/20240124222356/https://idl.cs.washington.edu/files/2021-FastKDE-VIS.pdf

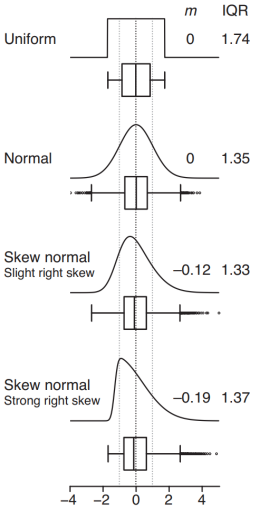
Boxplot I

1. Summarizes data into a '5-number' summary.
2. Detects extreme observation
3. The Centerline marks the median.
4. Box plots characterize a sample using the 25th, 50th, and 75th percentiles—also known as the lower quartile (Q1), median (m or Q2) and upper quartile (Q3)—and the interquartile range (IQR = Q3 – Q1), which covers the central 50% of the data.



But Why Quartiles? I

1. Quartiles are insensitive to outliers and preserve information about the center and spread.
2. They are preferred over the mean and variance for population distributions that are asymmetric or irregularly shaped and for samples with extreme outliers.
3. There is a special plot called **violin plot** that combines boxplot with kernel density estimation.



But Why Quartiles? II

Further Reading:

Visualizing samples with box plots,
https://web.archive.org/web/20240125000715/https://course.khoury.northeastern.edu/cs7280sp16/CS7280-Spring16_files/NatureMethods-boxplots.pdf

Plotting Libraries in Python I

1. Matplotlib
2. Seaborn: built on the top of Matplotlib, but comes with various color schemes and styles
3. Panda comes with its own functions that directly let you use Matplotlib
4. Plotly: interactive plots and dashboards; enables plotting in a web browser
5. plotly-resampler package lets use plotly for large datasets $> 1million$ datapoints
6. Some other dashboarding tools: Boken, Voila, Bowtie

The End