Lecture 1: Data Types and Descriptive Analysis

Statistical Methods for Data Science

Yinan Yu

Department of Computer Science and Engineering

October 30, 2023

Learning outcome

- Understand the four data types for statistics
- For each type, be able to compute descriptive statistics (in particular, sample mean, sample variance, frequency) and choose appropriate visualization tools
- Be able to compute histograms and quantiles from data
- Be able to identify other relevant visualization tools and justify their use cases

Data type

Data types for statistics

- Categorical data
  - Nominal data
  - Ordinal data
- Numerical data
  - Discrete (interval) data
  - Continuous (ratio) data

Categorical data

- Nominal data: labels or tags
  Example: the answer to the question "what types of ducks do you have at home?"
  - Scoter 
  - Goldeneye 
  - Domestic duck 
  - Wood duck 
  - King eider 

  Your answer can be stored as a list of *nominal data*, e.g. ["Goldeneyes", "Wood duck"].

Oops, now your personal data lives in the cloud. ChatGPT will probably train on it.

Categorical data

- Ordinal data: ordered labels or tags
  Example: the answer to the question "how much do you like wood ducks?"
  - Hate'em
  - Meh
  - Neutral
  - Yes
  - Super much! All my ducks are wood ducks!

  They are called *ordinal data* since they represent ordered categories.
  Note: they are ordered but there is no indication of the distance between two categories.

Numerical data

- Discrete (interval) data: values that are countable, e.g. $\mathbb{Z}$
  Example: the answer to the question "how many ducks do you have at home?" 20

Numerical data

- Continuous (ratio) data: values that are uncountable, e.g. $\mathbb{R}$. Example: the answer to the question "what is the weight of your favorite duck?" 4.5 kg

Now we know there are different types of data, let's get the analysis started!

But first, we need to put them into a *container* (data structure) so that we can easily manipulate them using a computer.

Data container

Data container

1. Array (tensor)
2. Table

Data container

1. Array (tensor):
    - Elements typically have the same numerical type
    - Elements are indexed by their locations
    - Dimension (order, rank) is the number of indices used to index each element

object	dimension	example
Scalar	0	0.1
Vector	1	$[0.1, 0.2, 3.5]$
Matrix	2	$\begin{bmatrix} 0.1, 0.2, 3.5 \\ 2.1, 0.8, 9.6 \end{bmatrix}$
Higher order tensor	≥ 3	$\left[\begin{bmatrix} 0.1, 0.2, 3.5 \\ 2.1, 0.8, 9.6 \end{bmatrix}, \begin{bmatrix} 8.4, 4.6, 5.7 \\ 1.9, 4.3, 2.8 \end{bmatrix}\right]$

Data container

2. Table:
    - Each column can have its own type
    - Typically indexed by column names and conditions on their values

duck name (Nominal)	pecking order (Ordinal)	age [yr] (Discrete)	weight [kg] (Continuous)
Tom	A	5	2.0
Jerry	B	12	1.2

Then you can query a value by, for example, "give me the name of the chonkest duck"

Some Python libraries for data container

```
import numpy as np # array (tensor)
import pandas as pd # tables
```

Some Python libraries for data container

- np.ndarray
  - Continuous numerical data

```
array([10.7, 10.6, 10.6, 10.6, 10.6, 10.7, 10.7, 10.6, 10.4, 10.3, 10.2,
       10. , 10. , 10. ,  9.7,  9.5,  9.3,  9.1,  8.9,  8.8,  8.8,  8.6,
        8.5,  8.3,  8.2,  8.1,  8.1,  8.3,  8.3,  8.1,  8.1,  8.1,  8. ,
        8. ,  7.8,  7.7,  7.7,  7.6,  7.7,  7.5,  7.6,  7.6,  7.5,  7.8,
        7.9,  8. ,  8.2,  8.4,  8.7,  8.8,  8.9,  9.2,  9.5,  9.9, 10.3,
       10.8, 11.3, 11.9, 12.3, 13.1, 13.6, 14.2, 14.9, 15.4, 15.9, 16.4,
       16.7, 17.2, 17.5, 17.8, 18. , 18.2, 18.4, 18.8, 19. , 19.2, 19.4,
       19.6, 19.8, 19.8, 20. , 20.2, 20.4, 20.2, 20.2, 20.4, 20.5, 20.7,
       20.8, 20.9, 21. , 21.3, 21.4, 21.6, 21.6, 21.5, 21.5, 21.6, 21.8])
```

  - Discrete numerical data

```
array([146, 146, 146, 145, 145, 144, 144, 144, 143, 143, 143, 142, 142,
       141, 141, 141, 141, 141, 141, 140, 140, 139, 139, 139, 139, 139,
       139, 139, 139, 139, 139, 139, 132, 132, 132, 131, 131, 130, 130,
       130, 131, 131, 130, 130, 130, 129, 129, 129, 125, 125, 125, 124,
       124, 123, 123, 123, 122, 122, 122, 121, 121, 120, 120, 120, 119,
       119, 118, 117, 116, 116, 115, 114, 117, 117, 117, 116, 116, 115,
       115, 115, 115, 115, 114, 114, 114, 113, 113, 113, 113, 113, 113,
       112, 112, 111, 111, 111, 110, 110, 109, 108], dtype=uint8)
```

- pd.DataFrame

	Survived	Pclass	Embarked	Sex
0	0	3	S	male
1	1	1	C	female
2	1	3	S	female
3	1	1	S	female
4	0	3	S	male

Recap: data types and containers

- Data type
  - Categorical data: labels, tags
    - Nominal data: not ordered labels
    - Ordinal data: ordered labels
  - Numerical data: numbers
    - Discrete (interval) data: countable values
    - Continuous (ratio) data: uncountable values
- Data container
  - Array (tensor)
    - numerical data type
    - Python container: numpy.ndarray
  - Table
    - mixed data type
    - Python container: pandas.DataFrame

Categorical data

Descriptive statistics - categorical data

- Count and compute the **frequency** of different labels
  Example: ask your ducks to stand in a row and look at the colors

duck id	1	2	3	4	5	6
color	green	red	blue	blue	blue	red

  What is the frequency of a duck being blue?

$$Count(color = "blue") = 3$$

$$Frequency(color = "blue") = 3/6 = 0.5$$

  As simple as that! But it is very useful! It is essentially how you estimate probabilities.
  Note: sometimes the words "frequency" and "count" are used interchangeably.

Descriptive statistics - categorical data

- Transformed into discrete numerical data, e.g. **one-hot encoding**

duck id	1	2	3	4	5	6
color	green	red	blue	blue	blue	red
one-hot	[0, 1, 0]	[1, 0, 0]	[0, 0, 1]	[0, 0, 1]	[0, 0, 1]	[1, 0, 0]

where we encode each color into a vector:

$$[bool(color == red), bool(color == green), bool(color == blue)]$$

Numerical data

Descriptive statistics - numerical data

Given a data set (a **sample**): $\{x_1, x_2, \cdots, x_N\}$, where x_i are scalars

- Centrality: "the position of the center"
  - sample mean: $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$

    $$\text{e.g.,}\quad \{0.8, 0.2, 2, 10, 6\} \rightarrow 3.8$$

  - median: sort x_i and median is the value in the middle

    $$\text{e.g.,}\quad \{0.8, 0.2, 2, 10, 6\} \rightarrow \{0.2, 0.8, 2, 6, 10\} \rightarrow 2$$

  - mode (discrete values): the most frequent value in a sample

    $$\text{e.g.,}\quad \{0, 0, 6, 6, 3, 3, 3\} \rightarrow 3$$

Descriptive statistics - numerical data

Given a data set (a **sample**): $\{x_1, x_2, \cdots, x_N\}$, where x_i are scalars

- Centrality: "the center position"
  - sample mean: $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} x_i$
  - median: sort x_i and median is the value in the middle
  - mode (discrete values): the most frequent value in a sample
- Dispersion: "the spread"
  - min, max, range: $\min\{x_i\}$, $\max\{x_i\}$, $\max\{x_i\} - \min\{x_i\}$
  - sample variance: $s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{\mathbf{x}})^2$
  - sample standard deviation: s
  - quantiles/percentiles: explained on page 46

Note: these are called **sample statics**

Descriptive statistics - numerical data

If you use a pandas.DataFrame container to store your data, the method describe() gives you a summary of descriptive statistics, e.g.

Example data

	sex	weight (kg)	height (cm)
0	Male	68.781904	162.310473
1	Male	74.110105	212.740856
2	Male	71.730978	220.042470
3	Male	69.881796	206.349801
4	Male	67.253016	152.212156
...	...	...	...

Descriptive statistics using pandas

	height	weight
count	9999.000000	9999.000000
mean	168.571702	73.224464
std	9.771363	14.560297
min	137.828359	29.347484
25%	161.303580	61.605559
50%	168.447465	73.119948
75%	175.697056	84.890898
max	200.656806	122.465267

Descriptive statistics - numerical data

- Centrality: "the center position"
- Dispersion: "the spread"
- Dependence: given a sample with paired values:

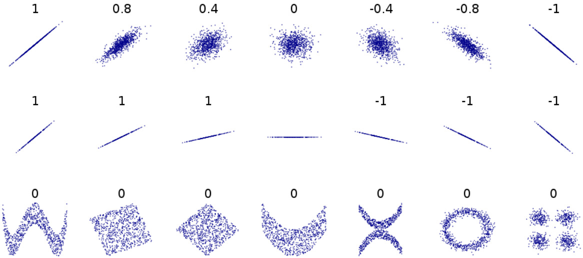
$$\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

  - covariance: $cov(x, y) = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})$
  - correlation: measures how close data is to a linear relationship

$$corr(x, y) = \frac{cov(x, y)}{s_x s_y}, \quad -1 \leq corr(x, y) \leq 1$$

Descriptive statistics - numerical data

Correlation example from Wikipedia:



Recap: descriptive statistics

- Categorical data
  - Count/frequency
  - Transformed into numerical, discrete data
- Numerical data
  - Centrality: mean, median, mode
  - Dispersion: min, max, range, quantiles/percentiles, variance/standard deviation
  - Dependence: covariance, correlation

Some Python libraries for visualization

```
import matplotlib.pyplot as plt
import seaborn as sns # more high level plotting functions
```

Categorical data

Categorical data example: titanic data

Wikipedia page: `https://en.wikipedia.org/wiki/Titanic`

Survived	Pclass	Embarked	Sex
0	3	S	male
1	1	C	female
1	3	S	female
1	1	S	female

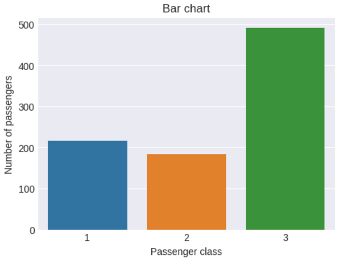
- Survived: if passenger has survived
- Pclass: passenger class (1: 1st; 2: 2nd; 3: 3rd)
- Embarked: port of embarkation (C: Cherbourg; Q: Queenstown; S: Southampton)
- Sex: passenger sex (male, female)

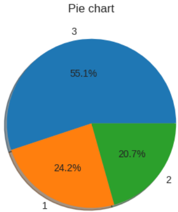
Visualization - categorical data

- Distribution
  - Bar chart
  - Pie chart
- Dependence
  - Mosaic plot

Distribution - bar chart vs pie chart

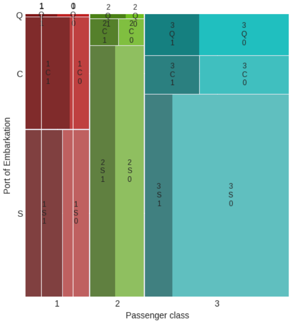
- Bar chart is usually preferred for
  - ordinal data
  - identifying differences
- Pie chart is used for visualizing the composition of things

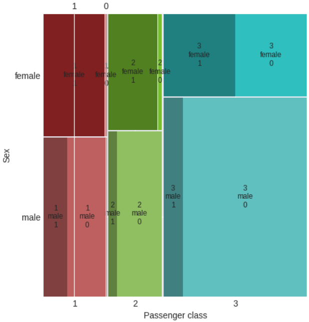




Dependence - mosaic plot

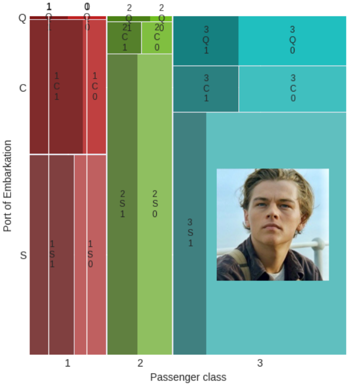
- Identify co-occurrences between multiple categorical variables
- Height and width represents the proportion of the corresponding category
- Too many variables in one plot can be confusing

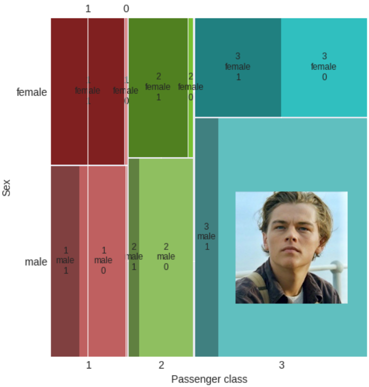




Dependence - mosaic plot

Jack didn't stand a chance!





Numerical data

Numerical data example: height and weight data

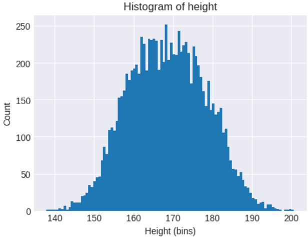
	sex	weight (kg)	height (cm)
0	Male	68.781904	162.310473
1	Male	74.110105	212.740856
2	Male	71.730978	220.042470
3	Male	69.881796	206.349801
4	Male	67.253016	152.212156

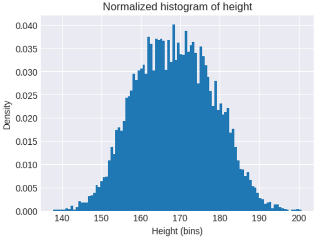
Visualization - numerical data

- Distribution:
  - Histogram
  - Normalized histogram
  - Kernel density estimator
  - Quantile/percentile
  - Box plot
- Dependence (two variables):
  - Scatter plot
  - Heat map for covariance/correlation

Distribution - histogram and normalized histogram

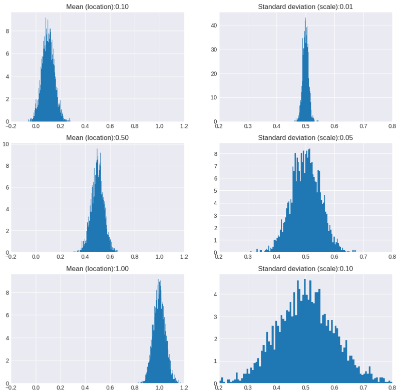
- Histogram:
  - Divide the range of the sample into equally sized bins
  - Count how many data points inside each bin
  - Plot the count (y-axis) vs bin values (x-axis)
- Normalized histogram: same as the histogram but the area is normalized to 1





Centrality vs dispersion

Example: left column - sample mean (location) vs right column - sample standard deviation (scale)



Distribution - kernel density estimator (KDE)

Kernel density estimator (KDE) is the smoothed normalized histogram.

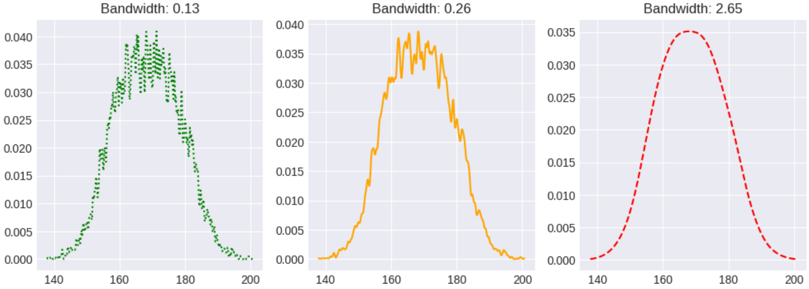
- Definition: given data set $\{x_1, x_2, \cdots, x_N\}$, KDE function is defined as

$$f_{KDE}(\mathbf{x}) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x_i - \mathbf{x}}{h}\right)$$

  where $K(\cdot)$ is a kernel function (you can find a bunch of them **here**); h is called the *bandwidth*; $\mathbf{x}$ is the *bin*.
- Intuition: think of it as a fancy *moving average* - hence the smoothing.

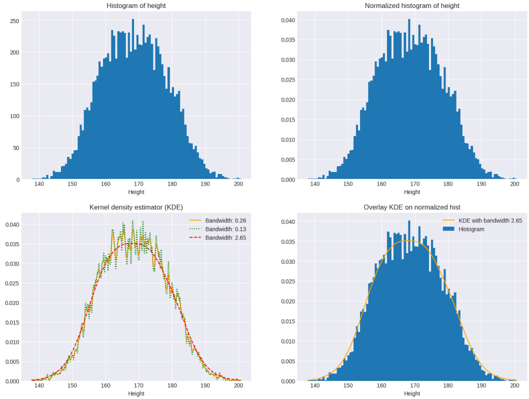
Distribution - kernel density estimator (KDE) (cont.)

Note: kernel function K and bandwidth h are *hyperparameters*. You choose them yourself and different choices will affect the outcome. For example, when we choose different bandwidths:



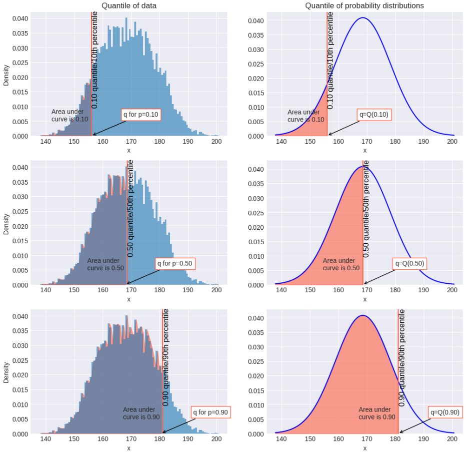
Recap

Histogram, normalized histogram, KDE with different bandwidths



Quantile/percentile

- Definition: given $p \in (0, 1)$, q is a p-quantile if $p \times 100\%$ of the data points are smaller than q.

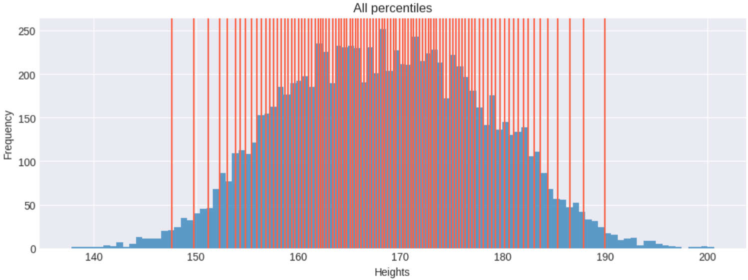


Quantile/percentile

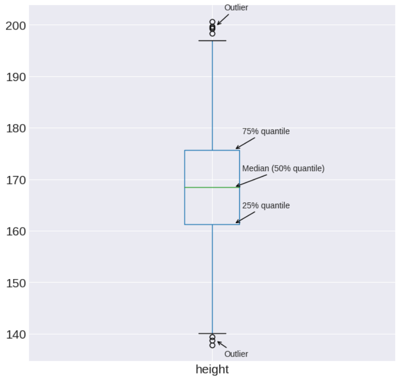
- How to compute quantile q: 1) sort the data from smallest to largest; 2) take the value q where $p \times 100\%$ of the data points are smaller than q.
  - Data: $[0, 7.3, 2, 1.5]$
  - Sorted: $[0, 1.5, 2, 7.3]$
  - 0.25-quantile: 1.5
  - 0.5-quantile: 2
  - 0.32-quantile: ? $\rightarrow$ need interpolation
  - Note: in practice, the result depends on the estimation and interpoltation methods
- In Python, it is calculated as np.quantile(data, p).
- Quantile/percentile can be calculated from either data or (spoiler alert) probability distributions.

Quantile/percentile

- Quantile and percentile are essentially the same, e.g. 0.3-quantile (alternatively 30%-quantile or 30th 100-quantiles) is the same as the 30th percentile.



Distribution - box plot

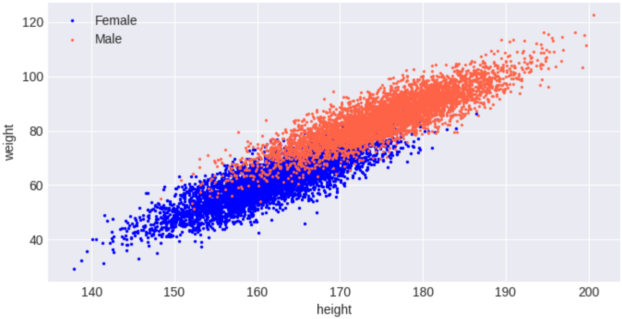


Dependence - scatter plot

Given a data set with two paired values:

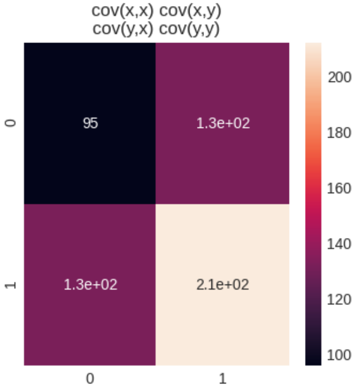
$$\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

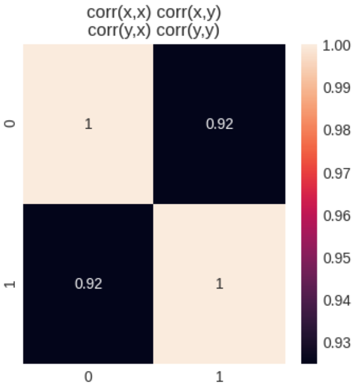
Two variables - variable y (weight) vs variable x (height)



Dependence - heat map

Covariance and correlation are defined on page 27





Summary

So far:

- Data types, data containers, descriptive statistics (e.g. sample mean, sample variance, data quantile), visualization (e.g. histogram)

Not yet:

- We can describe data we have seen, but we can't make predictions on unseen data.

Next:

- Probability distributions

Before next lecture:

- The data types we learned today
- The definition of histogram and how to compute them
- Be able to compute quantiles from data