Notes for STATS4710J Midterm RC Part 2

Peiran Wang

1 Visualization(1)

To build a better understanding for the target data for yourself and others.

1.1 Distribution

Denote a random variable as $X : \Omega \to \mathbb{R}$. And denote density function as $f_X : \mathbb{R} \to \mathbb{R}$ which can be used to represents the frequency of occurrence of the data (range of data).

For discrete random variables, density function can be defined as

$$f_X(x) = P[X = x].$$

For continuous random variables, the probability for any finite set will be 0 ($P[X = x] = 0 \ \ \forall x$). Then, the density function will be defined as

$$f_X(x) = \lim_{h \downarrow 0} P[X \in (x, x + h]]$$

if it exists.

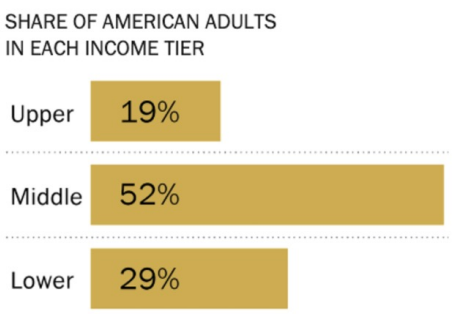
Note that for continuous random variables,

$$\int f_X(x)dx = 1 \quad \text{and} \quad f_X(x) \geq 0 \ \forall x$$

1.2 Bar Plot

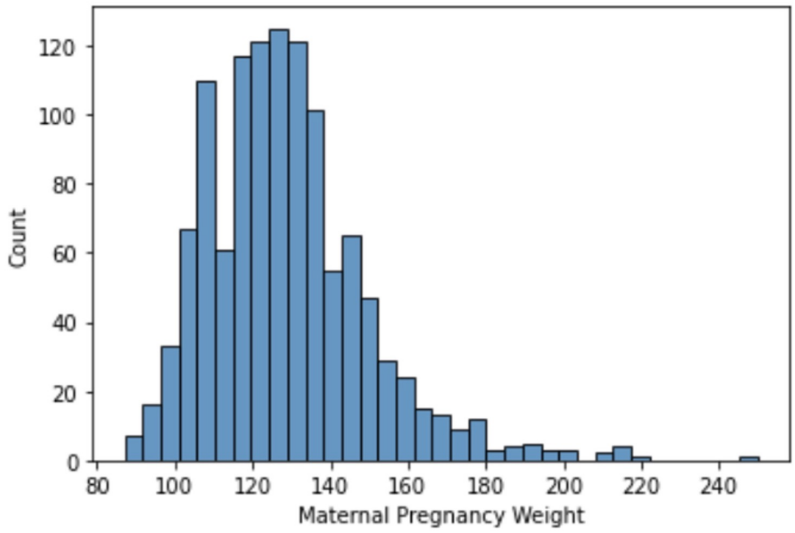
Bar Plots are the most common way of displaying the **distribution** of a **qualitative (categorical)** variable.

- For example, the proportion of adults in the upper, middle, and lower classes.
- **Lengths** encode **values**.
  - *Widths* encode *nothing*!
  - *Color* could indicate a sub-category (but not necessarily).



1.3 Histogram

For continuous variables, bar plot might not be able to show the pattern of the data. In this case, histogram can be used to estimate the density and represent the pattern of the data. Here is an example for histogram.



In this figure, the x-axis represents the value of the data, and the y-axis represents the counts of data in the given range.

1.3.1 Standardization

To better fit the properties of density function, $\int f_X(x)dx = 1$, the y-value will be modified to

$$y_i = \frac{n_i}{w_i * n}$$

where n_i is the original counting for i_{th} bin, w_i is the width for i_{th} bin and n is the total counting.

1.3.2 Mean

For a random variable X, if the density exists, the mean will be

$$\int x f_X(x)dx$$

Similarly, the mean for sample data will be its arithmetic mean.

1.3.3 Mode

It will be global maxima and local maxima.

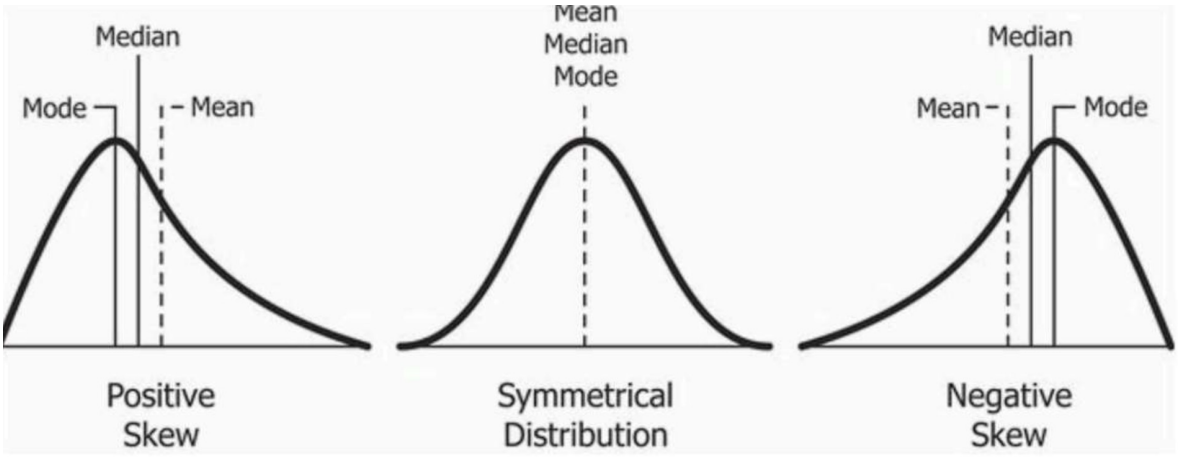
Note that it will be called unimodal if there's one mode, bimodal if there's two modes, multimodal otherwise.

1.3.4 Median

$$\int_{-\infty}^{median} f_X(x)dx = \int_{median}^{\infty} f_X(x)dx = \frac{1}{2}$$

1.4 Skewness

Right skewed (positive skewness):	Mode < Median < Mean
Left skewed (negative skewness):	Mean < Median < Mode



1.5 RDE

As histogram can only provide the step function estimation for the density. To obtain some smoother estimation, KDE (kernel density estimation) can be applied.

First, the kernel function will be applied to all the data points x_i. Kernel function can be any non-negative function with integration of 1. Commonly, Gaussian kernel will be used.

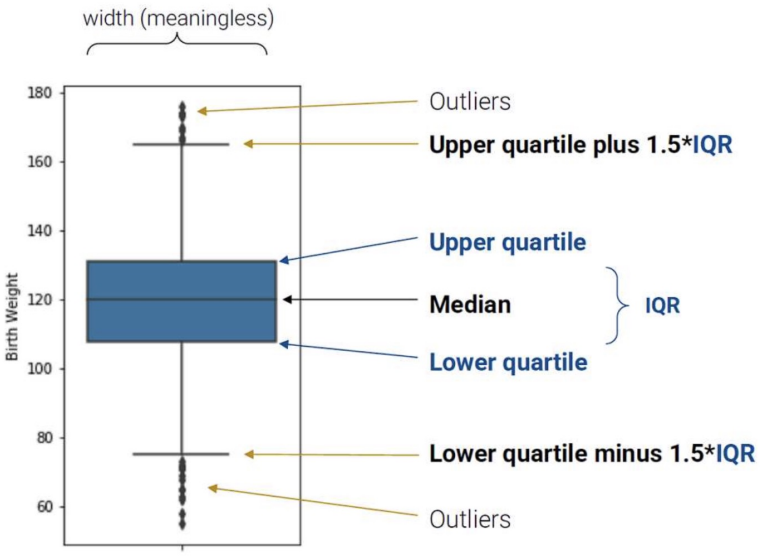
$$K_\alpha(x, x_i) = \frac{1}{\sqrt{2\pi}\alpha} \exp(-\frac{(x - x_i)^2}{2\alpha}).$$

Then, the final estimator will be the mean of such series of kernel functions

$$\hat{f}_\alpha(x) = \frac{1}{n} \sum_{i=1}^{n} K_\alpha(x, x_i)$$

Note that α is "bandwidth". Basically, the kernel $K_\alpha(x_i, x)$ has the mean x_i and variance α. Thus, the final estimator $\hat{f}_\alpha(x)$ will be smoother with larger "bandwidth".

1.6 Box Plot



Denote upper quartile as q3, lower quartile as q1.

$$\int_{\infty}^{q1} f_X(x)dx = \int_{q3}^{\infty} f_X(x)dx = 0.25$$

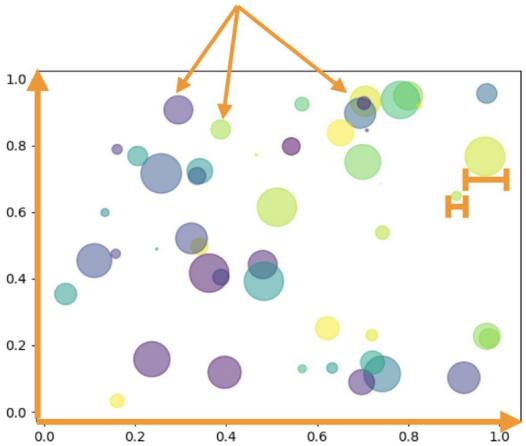
$$IQR = q3 - q1$$

2 Visualization(2)

2.1 Information Channel

How many variables are we encoding here?

- In other words, how many "channels" of information are there?



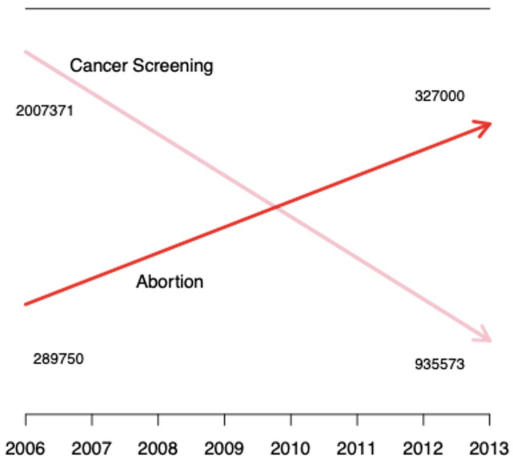
Answer: 4.

- x
- y
- area
- color

We could add even more: Shapes, outline colors of shapes, shading, etc. There are infinite possibilities.

2.2 Harnessing

2.2.1 Keep the Axis Consistent



The scales for the two lines are completely different!

- 327000 is smaller than 935573, but appears to be way bigger.
- **Do not use two different scales for the same axis!**

2.2.2 Color

- Qualitative: Choose a qualitative scheme that makes it easy to distinguish between categories. One category isn't "higher" or "lower" than another.
- Quantitative: Choose a color scheme that implies magnitude. Perceptually uniform colormaps have the property that if the data goes from 0.1 to 0.2, the perceptual change is the same as when the data goes from 0.8 to 0.9.

2.2.3 Marking

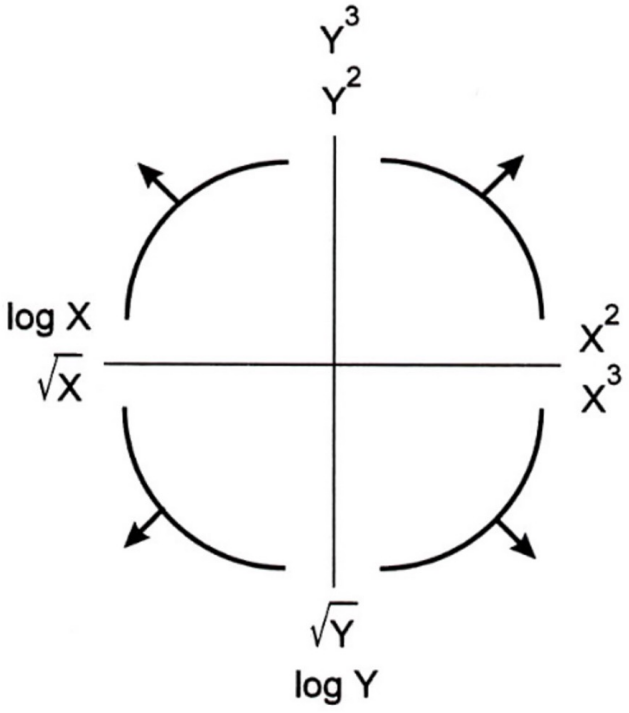
- Lengths are easy to distinguish; angles are hard
- Areas are hard to distinguish
- Avoid word clouds too! It's hard to tell the area taken up by a word.
- Avoid jiggling the baseline.

2.2.4 Context

- Informative title (takeaway, not description).
- Axis labels.
- Reference lines, markers, and labels for important values.
- Legends, if appropriate.
- Captions that describe the data.

2.3 Tukey-Mosteller Bulge Diagram

The Tukey-Mosteller Bulge Diagram is a guide to possible transforms to try to get linearity.



3 Linear Regression

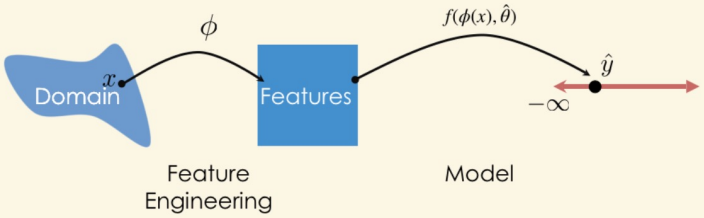
✅ 1. Choose a model	Multiple Linear Regression	$\hat{\mathbb{Y}} = \mathbb{X}\theta$
✅ 2. Choose a loss function	L2 Loss Mean Squared Error (MSE)	$R(\theta) = \frac{1}{n}\|\|\mathbb{Y} - \mathbb{X}\theta\|\|_2^2$
✅ 3. Fit the model	Minimize average loss with ~~calculus~~ geometry	$\hat{\theta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$
4. Evaluate model performance	Visualize, Multiple R^2	

Note that for simple linear model, design matrix $\mathbb{X}$ will be $n \times 2$ matrix where the first column is full of 1s and second column is the predictor.

4 Feature Engineering

The modeling process

The modeling process



1. Create (engineer) features to best reflect the "meaning" behind data.

2. Choose a model that is appropriate to capture the relationships between features (X) and the target/response (y).

3. Select a loss function and fit the model (i.e., determine θ).

4. Evaluate the model (e.g. using RMSE or R^2).

Some common transformation for data:

- One-Hot-Encoding
- Binarizer
- Standardization
- Transformation by Tukey-Mosteller Bulge Diagram

5 More about the Model

5.1 Properties for Expectation

For scalar a and b,

$$\mathbb{E}(aX + b) = a\mathbb{E}(X) + b.$$

For matrix A and vector b,

$$\mathbb{E}(AX + b) = A\mathbb{E}(X) + b.$$

For two random variables,

$$\mathbb{E}(X + Y) = \mathbb{E}(X) + \mathbb{E}(Y).$$

5.2 Properties for Variance

For scalar a and b,

$$Var(aX + b) = a^2 Var(X).$$

For Matrix A and vector b,

$$Var(AX) = AVar(X)A^T$$

For two random varibles,

$$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)$$

where

$$Cov(X, Y) = \mathbb{E}((X - \mathbb{E}(X))(Y - \mathbb{E}(Y)))$$

5.3 Bias Variance Trade-off

We've spent this section **defining** each component of the below equation.

$$\text{model risk} = \text{observation variance} + (\text{model bias})^2 + \text{model variance}$$

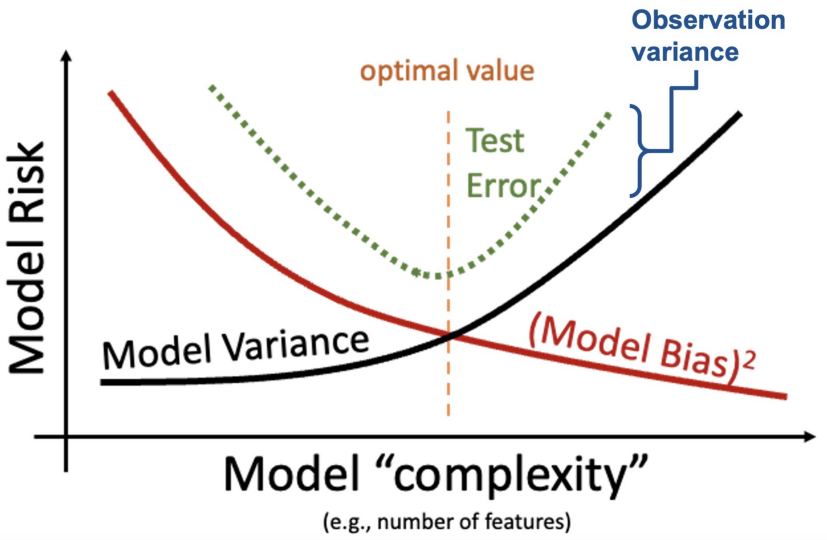
$$\mathbb{E}[(Y - \hat{Y}(x))^2] = \sigma^2 + \Big(\mathbb{E}[\hat{Y}(x)] - g(x)\Big)^2 + \text{Var}(\hat{Y}(x))$$

Interpret:

- Model risk is an expectation and is therefore a fixed number (for a given x and model $\hat{Y}(x)$).
- Observation variance is irreducible.
- As models **increase in complexity,** they **overfit** the sample data and will have **higher model variance**. This often corresponds to a decrease in bias.
- As models **decrease in complexity**, they **underfit** the sample data and have lower model variance. This corresponds to an **increase in bias**.

This is the **Bias-Variance Tradeoff**.

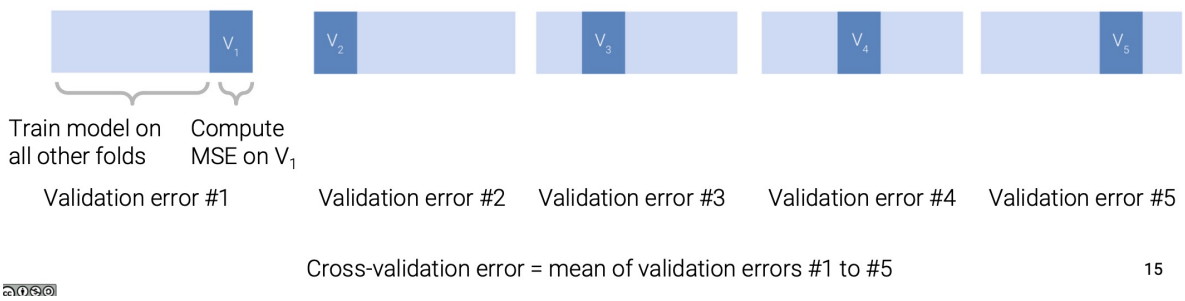
$$\text{model risk} = \text{observation variance} + (\text{model bias})^2 + \text{model variance}$$



6 Cross Validation

6.1 K-Fold Cross Validation

When there are several different model to choose (different type of model or models with different hyper-parameters), cross-validation can be applied to make a reasonable choice.



Actually, the cross-validation error is an estimate for the test error (mean square error on test data set. Therefore, the model with the smallest cross-validation error can be chose.

Also, note that commonly K can be 1 (simple hold-out validation method), 5, 10, N (leave-one-out-cross-validation).

6.2 Regularization

To decrease the risk of "overfit", some constrains can be applied on the estimator $\hat{\theta}$.

Intuitively, we want to control the complexity (ℓ_0 norm), the non-zero attributes of $\hat{\theta}$ under a certain limit. Unfortunately, this is a combinational problem and hard to solve in many cases.

6.2.1 ℓ_1 Regularization (LASSO)

This is the constrained optimization problem where the constrain is to control the ℓ_1 norm (sum of absolute value for all attributes) under a limit.

Generally, the results of this regularization will be smaller (closer to 0) than OLS results for all the attributes. Especially, some attributes will be decreased to 0.

6.2.2 ℓ_2 Regularization (Ridge)

This is the constrained optimization problem where the constrain is to control the ℓ_2 norm under a limit.

Generally, the results of this regularization will be smaller (closer to 0) than OLS results for all the attributes. Especially, it will not be decreased to 0 if it's not 0 for OLS.

6.3 Summary

Name	Model	Loss	Reg.	Objective	Solution
OLS	$\hat{\mathbb{Y}} = \mathbb{X}\theta$	Squared loss	None	$\frac{1}{n}\|\|\mathbb{Y} - \mathbb{X}\theta\|\|_2^2$	$\hat{\theta}_{\text{OLS}} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$
Ridge Regression	$\hat{\mathbb{Y}} = \mathbb{X}\theta$	Squared loss	L2	$\frac{1}{n}\|\|\mathbb{Y} - \mathbb{X}\theta\|\|_2^2 + \lambda\sum_{j=1}^{d}\theta_i^2$	$\hat{\theta}_{\text{ridge}} = (\mathbb{X}^T\mathbb{X} + n\lambda I)^{-1}\mathbb{X}^T\mathbb{Y}$
LASSO	$\hat{\mathbb{Y}} = \mathbb{X}\theta$	Squared loss	L1	$\frac{1}{n}\|\|\mathbb{Y} - \mathbb{X}\theta\|\|_2^2 + \lambda\sum_{j=1}^{d}\|\theta_i\|$	**No closed form**

Note that there is a hyper-parameter λ for regularization. And it can be determined by cross-validation method.

7 SQL

7.1 Genral Expression

```
SELECT <column expression list>
FROM <table>
[WHERE <predicate>]
[GROUP BY <column list>]
[HAVING <predicate>]
[ORDER BY <column list>]
[LIMIT <number of rows>]
[OFFSET <number of rows>];
```

- By convention, use **all caps** for keywords in SQL statements.
- Use **newlines** to make SQL code more readable.
- **AS** keyword: rename columns during selection process.
- **WHERE: rows; HAVING: groups. WHERE precedes HAVING.**

7.2 EDA

- Working with Text: 'LIKE'
- Converting Data Types: 'CAST'
- Applying Conditions: 'CASE'

7.3 Join

7.3.1 Cross Join

In a cross join, we find every possible combination of rows across the two tables. A cross join is also called a cartesian product.

7.3.2 Inner Join

Conceptually, you can imagine an inner join as a cross join filtered to include only matching rows.

7.3.3 Left Outer Join

In a left outer join (or just left join), keep all rows from the left table and only matching rows from the right table. Fill NULL for any missing values.

7.3.4 Right Outer Join

In a right outer join (or just right join), keep all rows from the right table and only matching rows from the right table. Fill NULL for any missing values.

7.3.5 Full Outer Join

In a full outer join, keep all rows from both the left and right tables. Pair any matching rows, then fill missing values with NULL. Conceptually similar to performing both left and right joins.

8 Reference

Mid RC SP2024, Youchen Qing Lecture Slide SU2024, Ailin Zhang

9 Appendix

The annotated draft of mid rc is attached in the end of the document.

Visualization

- distribution
- bar plot
- histogram
- KDE
- box plot

- Information channels
- Harnessing
  - x / y
  - color
  - marking
  - conditioning
  - texting

- Distribution

  Random Variable $X : \Omega \to \mathbb{R}$

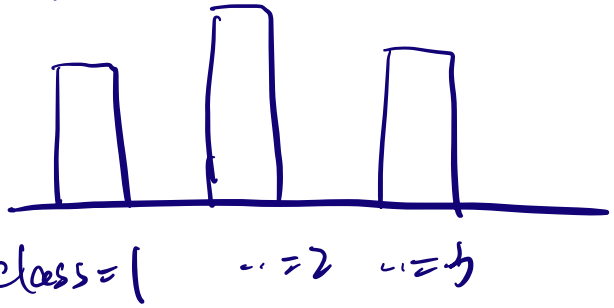
  - discrete RV: density $f_X(x) = P[X = x]$
  - continuous RV: $P[X = x] = 0 \quad \forall x \in \mathbb{R}$

$$F_X(x) = P[X \in (-\infty, x]]$$

$$\Rightarrow \text{density} = f_X(x) = F_X'(x)$$

$$\left(\lim_{x \to x} P[X \to x] \right)$$

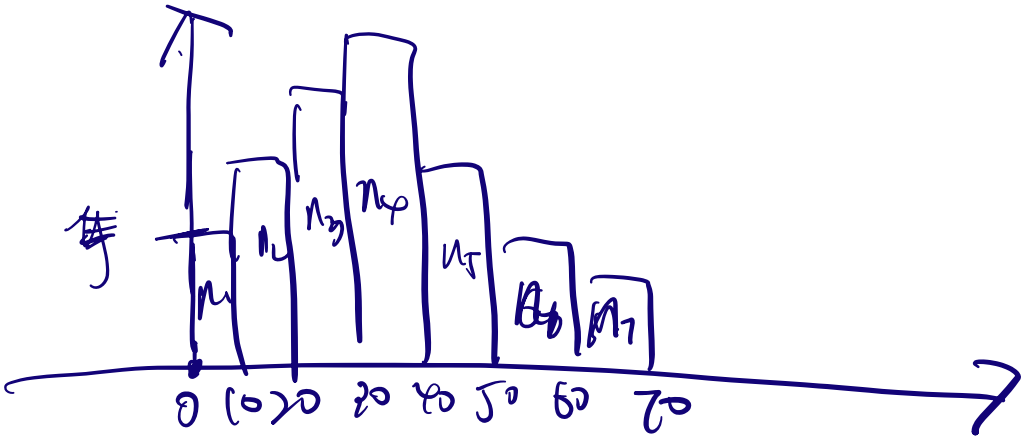
Barplot



$$f_X(x) \geq 0$$

$$\int_{-\infty}^{\infty} f_X(x) = 1$$

Histogram.



$$\sum w_i \cdot n_i$$

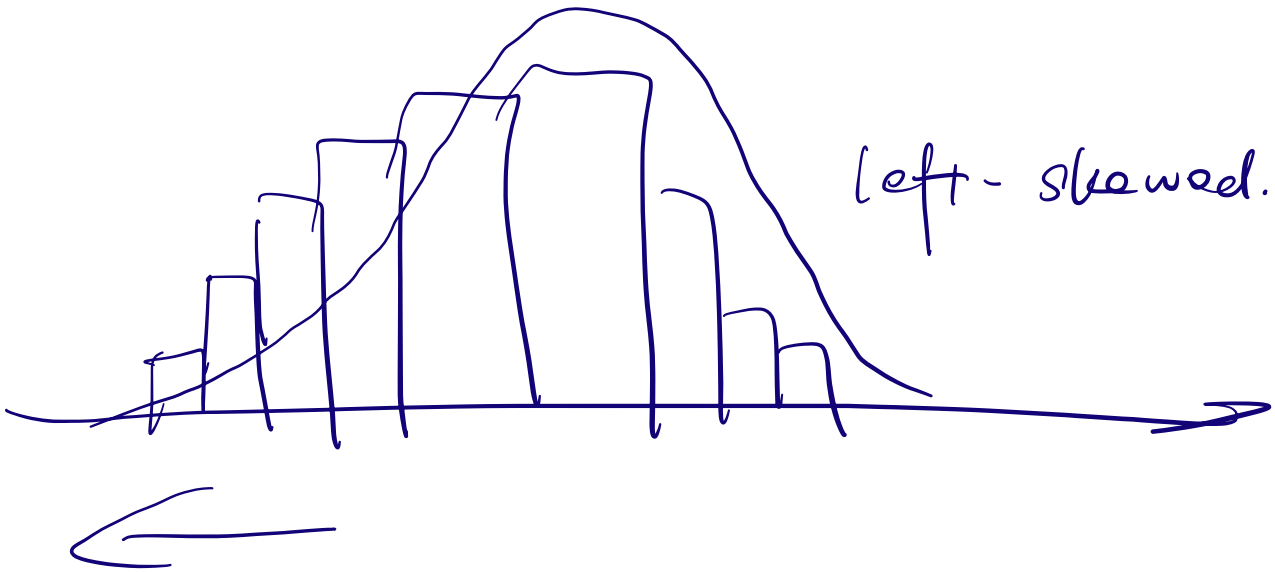
$$\sum w_i \cdot y_i = 1$$

$$w_i \cdot y_i = \frac{n_i}{\sum w_i \cdot n_i} \rightarrow N$$

mean = $\int_{-\infty}^{\infty} x f(x)$

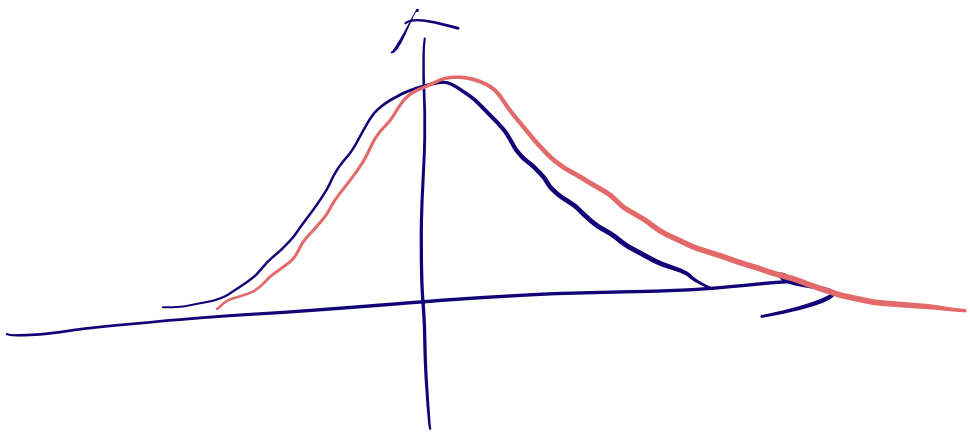
mode = local or global maximum

median = M $\int_{-\infty}^{M} f(x) = \frac{1}{2}$



right - skewed :

mode < median < mean



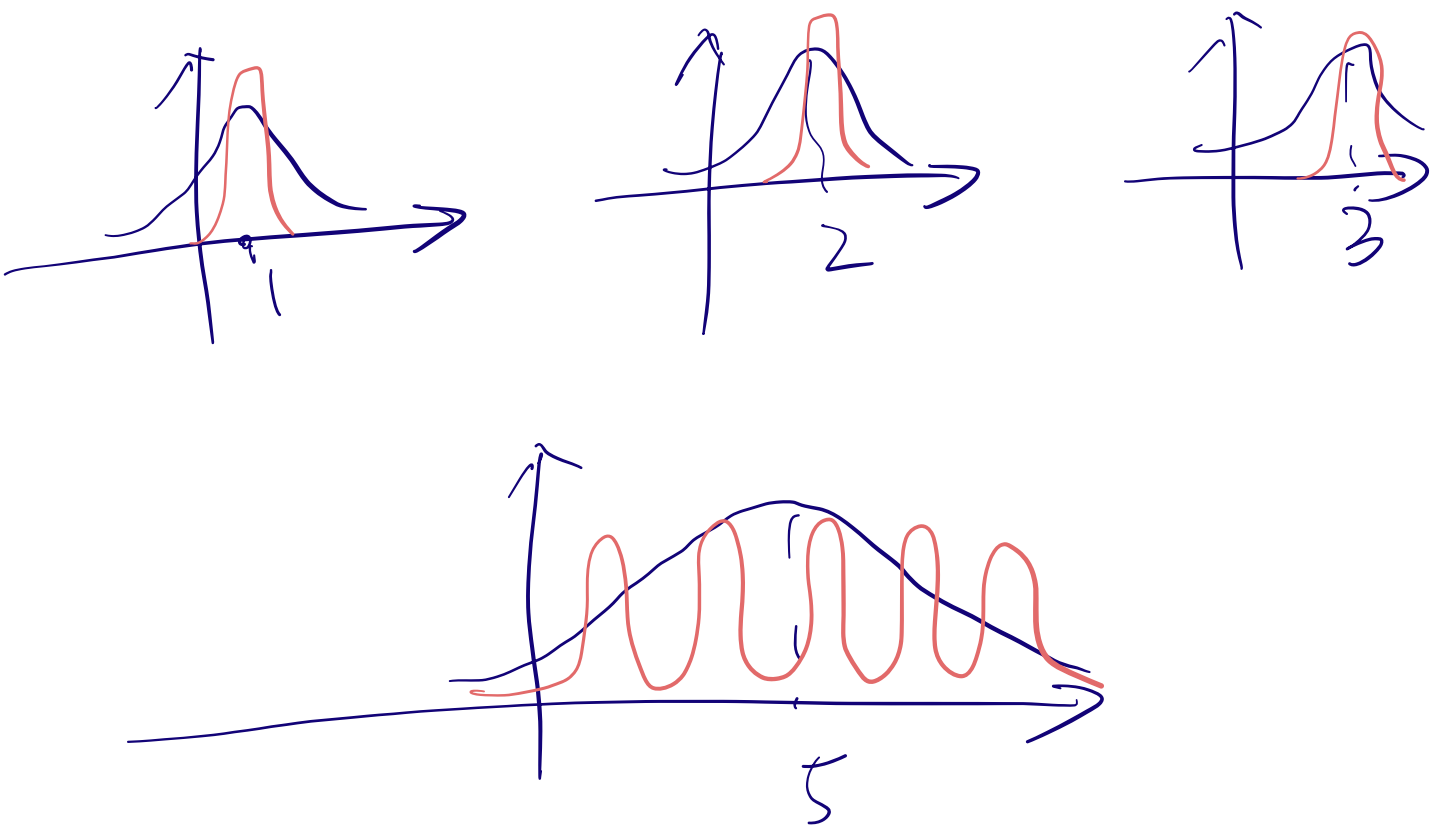
KDE.

$A = [a_1, \ldots, a_n]$

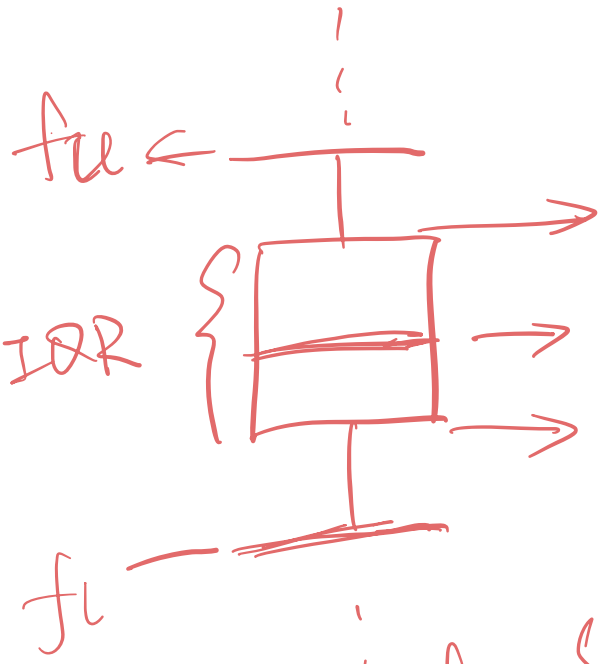
$$f_i(x) = \frac{1}{\sqrt{2\pi}\,\alpha} e^{\frac{(x-a_i)}{2\alpha^2}}$$

bandwidth

$$\frac{1}{n}\sum_i^n f_i(x) \leftarrow \text{kde}.$$



$$\int_{-\infty}^{\text{upper q}} f(x) = 0.75$$



near fence $\begin{cases} f_l = \text{median} - 1.5\ \text{IQR} \\ f_u = \text{median} + 1.5\ \text{IQR} \end{cases}$

outer fence $\begin{cases} F_{\cdot} = \text{median} \pm 3\,\text{IQR} \end{cases}$

Mid RC for STAT4710J

by Youchen Qing 2024/4/1 Lecture 10-18

Visualization (Lecture 10-11)

GOAL:

1. To help your own understanding of your data/results.
2. To communicate results/conclusions to others.

Distribution

Probability distribution or probability density function is a function f_X associated to a discrete random variable X defined as

$f_X : \Omega \to \mathbb{R}$

with Ω a countable subset of $\mathbb{R}$ satisfying the properties that

(i) $f_X(x) \geq 0 \, \forall x \in \Omega$ and

(ii) $\sum_{x \in \Omega} f_X(x) = 1.$

For a continuous random variable X the probability distribution is defined as

$f_X : \mathbb{R} \to \mathbb{R}$

with the properties that

(i) $f_X(x) \geq 0 \, \forall x \in \mathbb{R}$ and

(ii) $\int_{x \in \mathbb{R}} f_X(x) \, dx = 1.$

Bar Plot

A **Bar plot** shows the relationship between a numeric and a categorical variable, with:

- Entities represented as bars.
- Values represented as the size of the bars.

In Seaborn

For plotting the number of occurrences in each category:

```
seaborn.countplot(data=None, x=None, y=None, hue=None)
```

For a general bar plot:

```
seaborn.barplot(data=None, x=None, y=None, hue=None, estimator='mean')
```

The hue argument in Seaborn is used for colour encoding.

Histogram

Vertical bar graph

- Each bar represents the proportion or number of data in a given range
- Categories are called bins

Skewness and tail

- Tail on the left—left skewed
- Tail on the right—right skewed

Mode

- Local or global maximum
- Number mostly depends on the density curve (KDE)

$n = w \times h \times N$

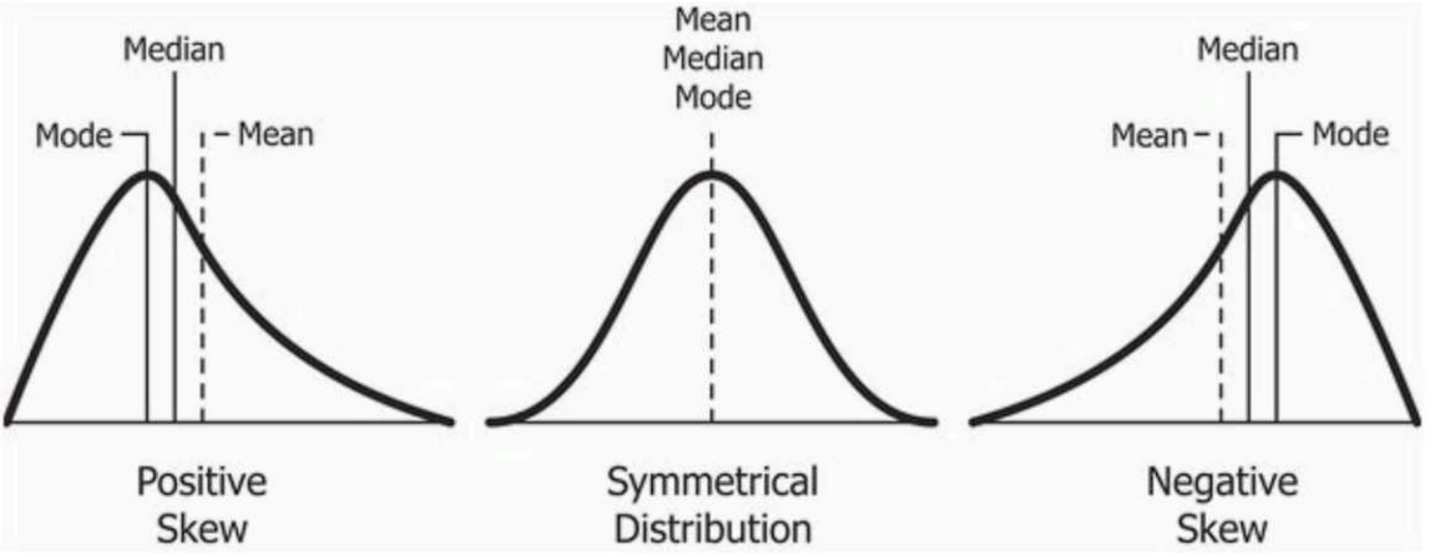
- n : number of samples in a bin
- w : bin width
- h : bar height
- N : total number of samples

Right skewed (positive skewness)

- Mode < Median < Mean

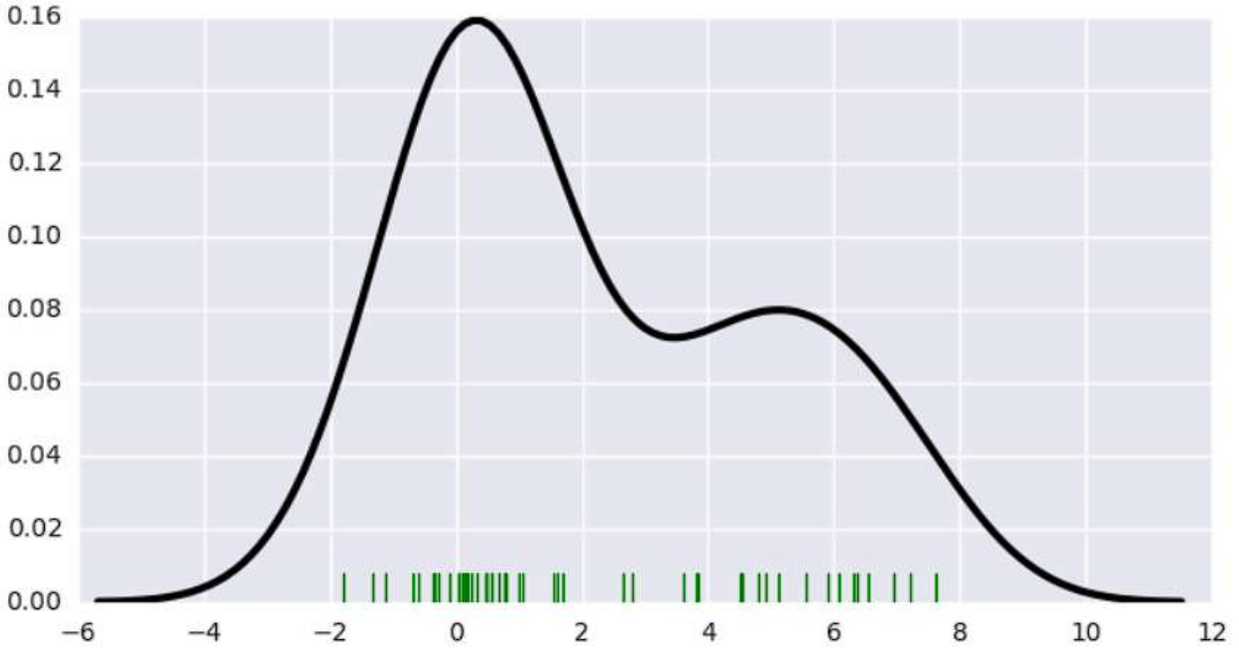
Left skewed (negative skewness)

- Mean < Median < Mode



Rug plot

- Displayed as marks along an axis
- Could be seen as 1-D scatter plot



KDE

A kernel density estimation (KDE) is used to estimate a probability density function (distribution).

Three steps

1. Place a kernel at each data point
2. Normalize kernels
    - Divide each kernel by the number of kernels in total
3. Sum kernels

Quartiles

- 25% of the data are no greater than the first quartile q_1
- 50% of the data are no greater than the second quartile q_2
- 75% of the data are no greater than the third quartile q_3

Interquartile range (IQR)

$IQR = q_3 - q_1$

Box plot

Fences

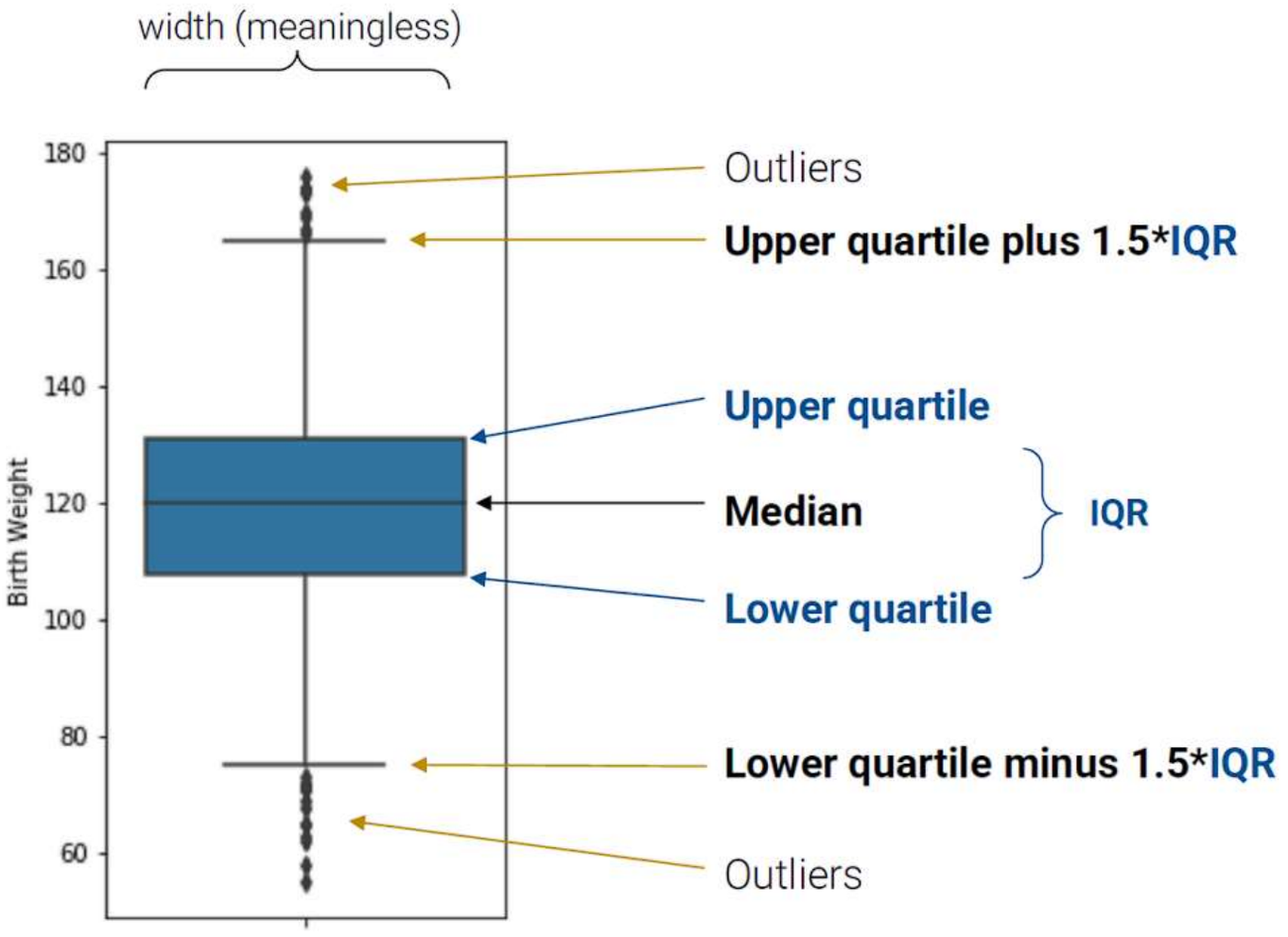
- Inner fences
  - $f_1 = q_1 - 1.5 \times IQR$
  - $f_3 = q_3 + 1.5 \times IQR$
- Outer fences
  - $F_1 = q_1 - 3 \times IQR$
  - $F_3 = q_3 + 3 \times IQR$

Adjacent values (where the line extending to the left and right of the box ends)

- $a_1 = \min\{x_k : x_k \geq f_1\}$
- $a_3 = \max\{x_k : x_k \leq f_3\}$

Outliers

- Near outliers: outside the inner fences but inside the outer fences
- Far outliers: outside the outer fences



Choosing methods of visualization accordingly

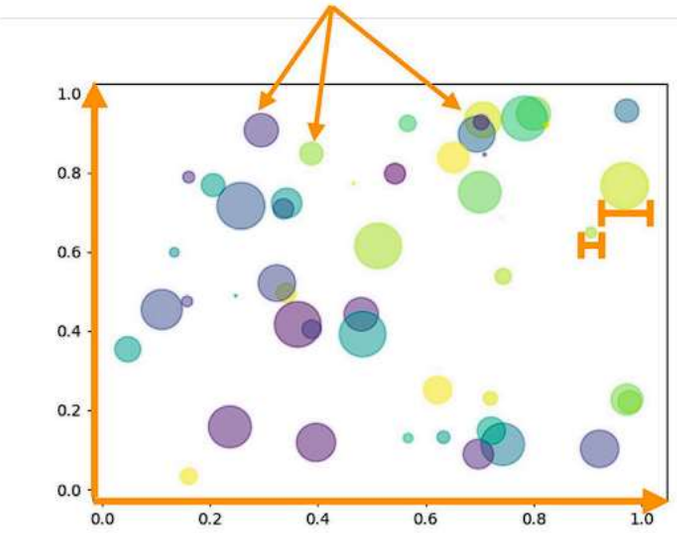
- Comparing quantitative distributions:
  - overlaid histograms and density curves (unclearness vs. completeness)
  - side by side box/violin plots
- Relationships between quantitative variables:
  - Previously (histogram: number/frequency of value)
  - scatter plot: relationship between pairs of numeric variables
  - scatter plot suffers overplotting
    - add some random noise to x variable
    - adjust the transparency (alpha)
  - hex plot:
    - why use hexagons instead of squares?
    - marginal distribution are shown as histograms
  - contour plot:
    - 2 dimensional version of density curves
    - marginal distribution: density curve
- Interactive data visualization (for the web)

Visualization Theory

- **Information channels**

How many variables are we encoding here?

- In other words, how many "channels" of information are there?



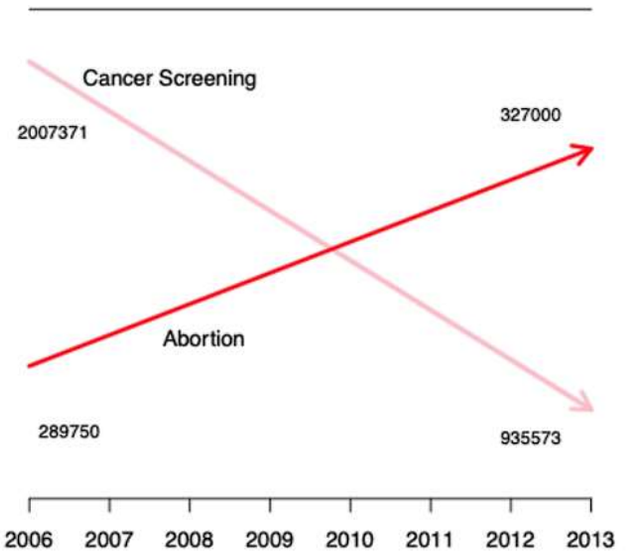
Answer: 4.

- x
- y
- area
- color

We could add even more: Shapes, outline colors of shapes, shading, etc. There are infinite possibilities.

- **Axis**

Keep axis scales consistent (try to choose axis limits to fill the visualization)

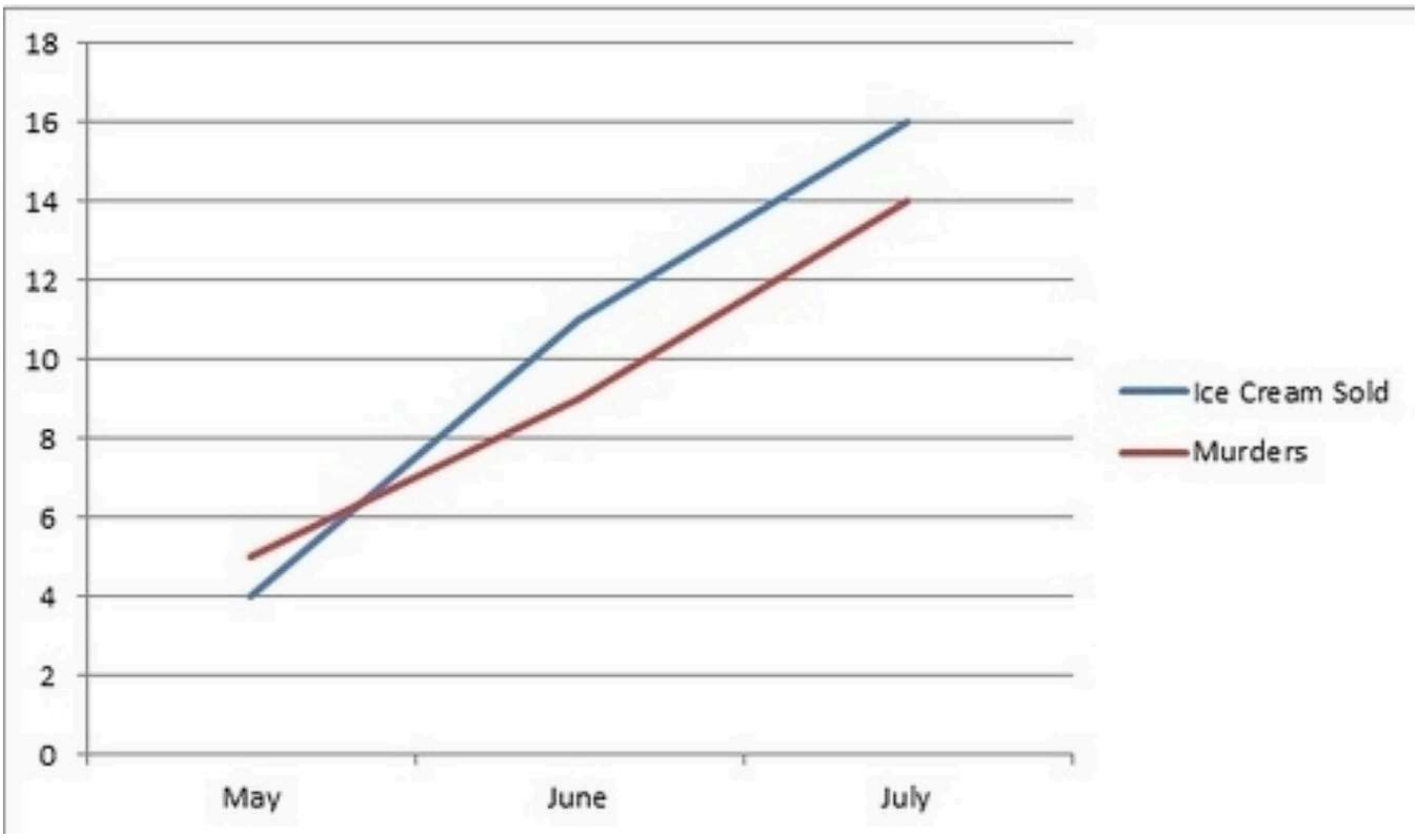


The scales for the two lines are completely different!

- 327000 is smaller than 935573, but appears to be way bigger.
- **Do not use two different scales for the same axis!**

- **Color**
- Perceptually uniform colormaps have the property that if the data goes from 0.1 to 0.2, the perceptual change is the same as when the data goes from 0.8 to 0.9.
- Qualitative:
  - Choose a qualitative scheme that makes it easy to distinguish between categories.
  - One category isn't "higher" or "lower" than another.
- Quantitative: Choose a color scheme that implies magnitude
- **Markings**
  - Lengths are easy to distinguish; angles are hard

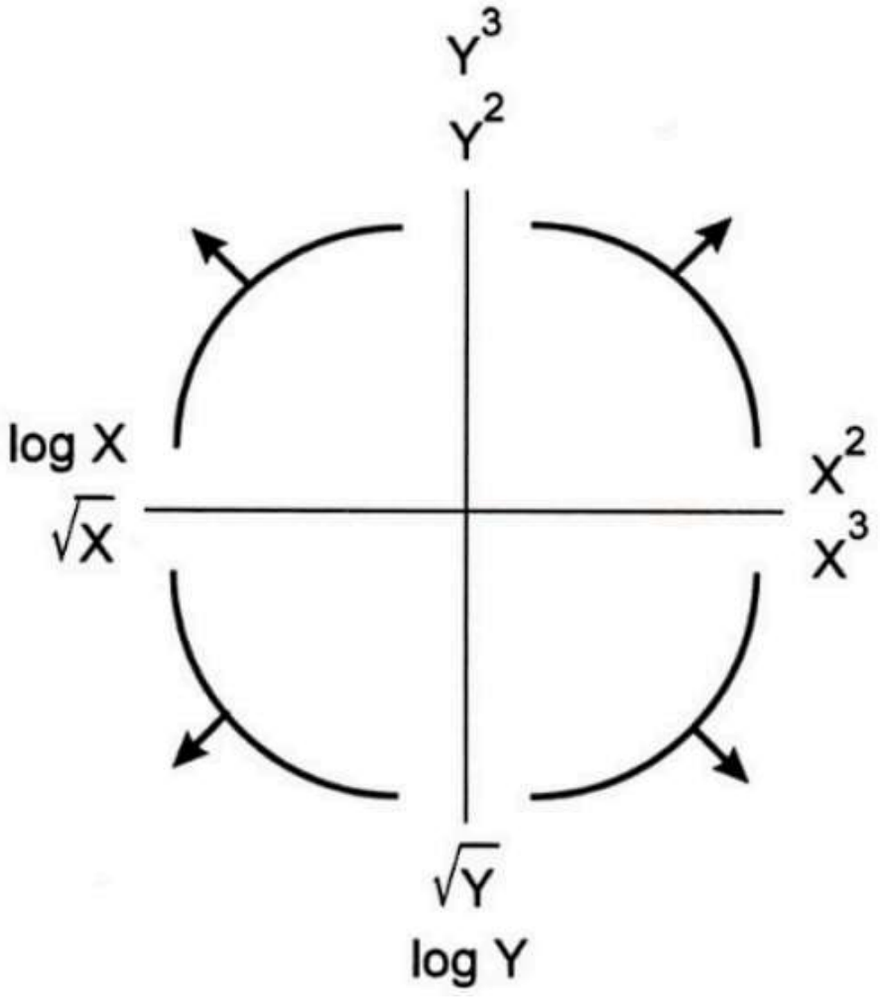
    - Avoid jiggling the baselines
- **Give informative titles, axis, labels, legends**
- **The plots make sense**



Tukey's Bulging Rule

Idea: Transform the values on the axes to get a (almost) linear relation.

General strategy:



The diagram illustrates how different transformations applied to the X and Y axes can linearize different types of relationships between variables. This strategy is used to identify the right transformation for achieving linearity in regression analysis.

Modeling

Loosely speaking, a (machine learning) model is a function *f* defined as

$f : A \rightarrow B$

with *A* the feature space and *B* the target space.

Feature space: Variables you use to train the model
Target space: Variables you want to study

In machine learning, all we do is to find a f that makes most sense.

Feature Engineering

- Select features from the feature space
- Create new features that are not originally in the feature space
- Enhance model performance

Encoding

Ways to encode categorical variables

- Label Encoding
- One-Hot Encoding
- Hash Encoding
- Target Encoding

Ways to encode words

- Bag-of-words encoding
  - Based on measuring the similarity between vectors
- TF-IDF
  - Take into account the importance of words

Choose the strategy that makes most sense.

Loss function and objective function

A loss function *L* is defined as

$L : T \rightarrow \mathbb{R}$

with *T* the tuple space of (y, ŷ).

- y: ground truth value
- ŷ: predicted value

An objective function is either a loss function or its opposite. We want to minimise (or maximise) the objective function to get the optimal model.

Simple linear regression (SLR)

- Model: $\hat{y} = a + bx$
- Loss function: $L(y, \hat{y}) = (y - \hat{y})^2$
- Objective function: $R(\theta) = \frac{1}{n} \sum_{i=1}^{n} L(y^{(i)}, \hat{y}^{(i)})$
- Optimal solution:
  - $\hat{b} = r \frac{\sigma_y}{\sigma_x}$
  - $\hat{a} = \bar{y} - \hat{b}\bar{x}$

r is the correlation,
$r = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{x^{(i)} - \bar{x}}{\sigma_x} \right) \left(\frac{y^{(i)} - \bar{y}}{\sigma_y} \right)$

Properties

- Passes $(\bar{x}, \bar{y})$

- Residuals sum up to 0

Constant model

- Model: $\hat{y} = \theta$
- Loss function: mean squared error (MSE) or mean absolute error (MAE)
- Optimal solution:
  - $\hat{\theta} = \text{mean}(y)$ for MSE
  - $\hat{\theta} = \text{median}(y)$ for MAE

Pros and cons

- With MSE:
  - Pros: differentiable; Easy to find the optimum
  - Cons: sensitive to outliers
- With MAE:
  - Pros: robust to outliers
  - Cons: piece-wise function; hard to find the optimum

General linear regression

Linear least squares

- Model: $\hat{y} = \sum_{j=1}^{m} \theta_j x_j$ (assuming $x_{[1]} = 1$, which is the bias term)
- Loss function: MSE
- Optimal solution: $\hat{\theta} = (X^T X)^{-1} X^T Y$

Properties:

- Residuals sum up to 0 (only when there is a bias term)
- $\hat{\theta}$ is unique $\iff$ X is column full rank

Keep the sample space larger than the feature space.

OLS derivation

Optimisation problem

minimize over θ:
$\min_\theta \|Y - X\theta\|_2$

The loss function is
$L(\theta) = \|Y - X\theta\|_2$
$= (Y - X\theta)^T (Y - X\theta)$
$= Y^T Y - Y^T X\theta - \theta^T X^T Y + \theta^T X^T X$

Set the gradient of L to 0
$\frac{\partial L(\theta)}{\partial \theta} = \frac{\partial (Y^T Y - Y^T X\theta - \theta^T X^T Y + \theta^T X^T X)}{\partial \theta}$
$= -2X^T Y + 2X^T X\theta = 0$

Hence,
$\hat{\theta} = (X^T X)^{-1} X^T Y$

You can refer to the slides for linear operations

Logistic Regression

Optimization problem

The goal is to find the best parameters θ that minimize the logistic regression cost function, which is the binary cross-entropy loss for a dataset with m examples:

$\min_\theta J(\theta)$

The loss function $J(\theta)$ for logistic regression is defined as:

$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right]$

where $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ is the hypothesis function for logistic regression.

The gradient of the loss function with respect to θ is:

$\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{m} X^T (h_\theta(X) - Y)$

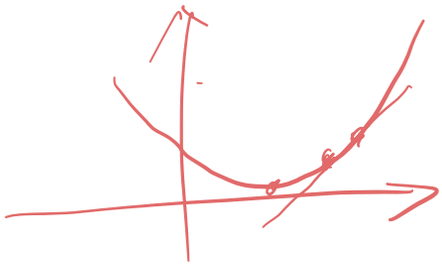
Where:

- X is the matrix of input features,
- Y is the vector of observed outputs (0 or 1),
- $h_\theta(X)$ is the vector of predicted probabilities.

The parameters θ are updated iteratively using an optimization algorithm such as gradient descent:

$\theta := \theta - \alpha \frac{\partial J(\theta)}{\partial \theta}$

where α is the learning rate.



Properties:

- Output is a probability that the given input point belongs to the class labeled as "1".
- Logistic regression models a binary outcome.

Cross Validation Summary

When selecting between models, we want to pick the one that we believe would generalize best on unseen data. Generalization is estimated with a "cross validation score". When selecting between models, keep the model with the best **score**.

Two techniques to compute a "cross validation score":

- The Holdout Method: Break data into a separate **training set** and **validation set**.
  - Use **training set** to fit parameters (thetas) for the model.
  - Use **validation set** to score the model.
  - Also called "Simple Cross Validation" in some sources.
- K-Fold Cross Validation: Break data into K contiguous non-overlapping "folds".
  - Perform K rounds of Simple Cross Validation, except:
    - Each fold gets to be the **validation set** exactly once.
    - The final **score** of a model is the average validation score across the K trials.

SQL

```
SELECT [DISTINCT] <column expression list>
FROM <table>
[WHERE <predicate>]
[GROUP BY <column list>]
[HAVING <predicate>]
[ORDER BY <column list>]
[LIMIT <number of rows>]
[OFFSET <number of rows>];
```

Note: Column Expressions may include aggregation functions (MAX, MIN, etc) and DISTINCT.

SQL Joins

INNER JOIN

An `INNER JOIN` retrieves records that have matching values in both tables. It returns rows when there is at least one match in both tables, excluding rows with no match.

FULL OUTER JOIN

A `FULL OUTER JOIN` returns all records when there is a match in the left, right, or both tables. It combines the results of both `LEFT` and `RIGHT OUTER JOIN`.

CROSS JOIN

A `CROSS JOIN` produces the Cartesian product of two tables, combining each row from the first table with each row from the second table.

LEFT OUTER JOIN (or LEFT JOIN)

A `LEFT OUTER JOIN` returns all records from the left table and the matched records from the right table. If there is no match, the result from the right side will be `NULL`.

RIGHT OUTER JOIN (or RIGHT JOIN)

A `RIGHT OUTER JOIN` returns all records from the right table and the matched records from the left table. If there is no match, the result from the left side will be `NULL`.

Reference

Mid_RC_Part2 Yuxuan Zheng
Mid_RC_note Sizhe Zhou
lecture slides 2023 summer