CSCI 347, Data Mining

Evaluation – 2 topics:

1. Holdout Methods
2. Confidence Levels

Testing

Testing is necessary because data mining methods do not adhere to the assumptions required by statistical theory. Typically holdout data for testing.

Train, Validation, Test Split

If enough data:

- Split data - training and testing (80/20 is a good starting point)
- Split *training* data - training and validation (80/20 is a fair split)
- Subsample random selections of your training data, train the classifier with this, and record the performance on the validation set
- Try a series of runs with different amounts of training data:

  randomly sample 20% 10 times and observe performance on the validation data, then do the same with 40%, 60%, 80%. You should see both greater performance with more data, but also lower variance across the different random samples
- To get a handle on variance due to the size of test data, perform the same procedure in reverse. T

  Train on all of your training data, then randomly sample a percentage of your *validation* data a number of times, and observe performance. You should now find that the mean performance on small samples of your validation data is roughly the same as the performance on all the validation data, but the variance is much higher with smaller numbers of test samples

https://stackoverflow.com/questions/13610074/is-there-a-rule-of-thumb-for-how-to-divide-a-dataset-into-training-and-validatio

Holdout Methods

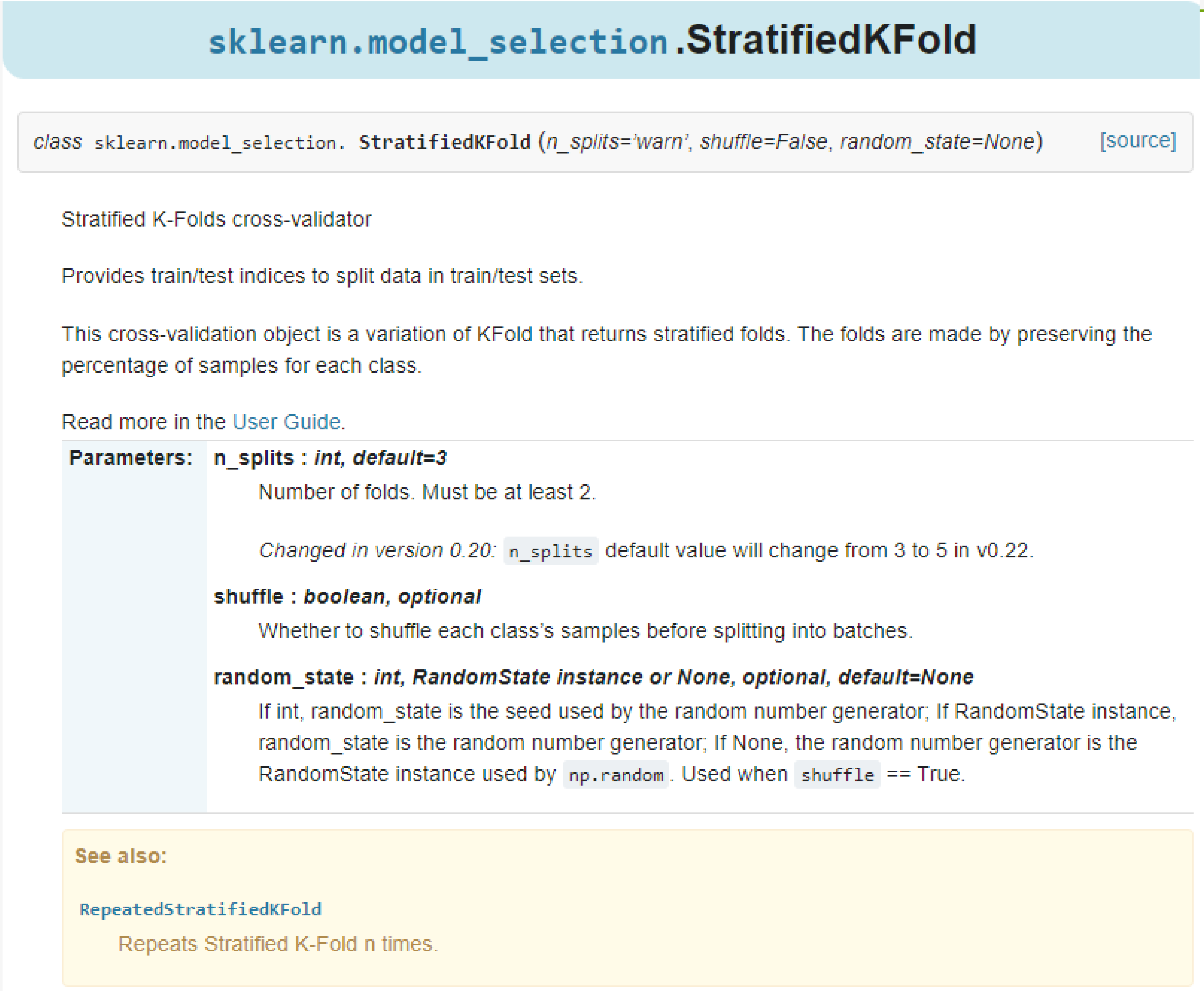
Stratified holdout –represent each class in approximately equal proportions in the testing dataset as it was in the overall dataset

Repeated holdout method – Use multiple iterations, in each iteration randomly select a certain proportion of the dataset for training (possibly with stratification). Average the error rates on the different iterations to yield an overall error rate

Cross-Validation

Cross-validation – decide a fixed number of "folds" or partitions of the dataset. For each of the n folds, train with (n-1)/n of the dataset, test with 1/n of the dataset. Average the resulting n error estimations.

10-fold cross validation with stratification has become the standard.



sklearn.model_selection.StratifiedKFold

class `sklearn.model_selection.` **`StratifiedKFold`**(*n_splits='warn', shuffle=False, random_state=None*) [source]

Stratified K-Folds cross-validator

Provides train/test indices to split data in train/test sets.

This cross-validation object is a variation of KFold that returns stratified folds. The folds are made by preserving the percentage of samples for each class.

Read more in the User Guide.

Parameters:

n_splits : *int, default=3*

Number of folds. Must be at least 2.

Changed in version 0.20: `n_splits` default value will change from 3 to 5 in v0.22.

shuffle : *boolean, optional*

Whether to shuffle each class's samples before splitting into batches.

random_state : *int, RandomState instance or None, optional, default=None*

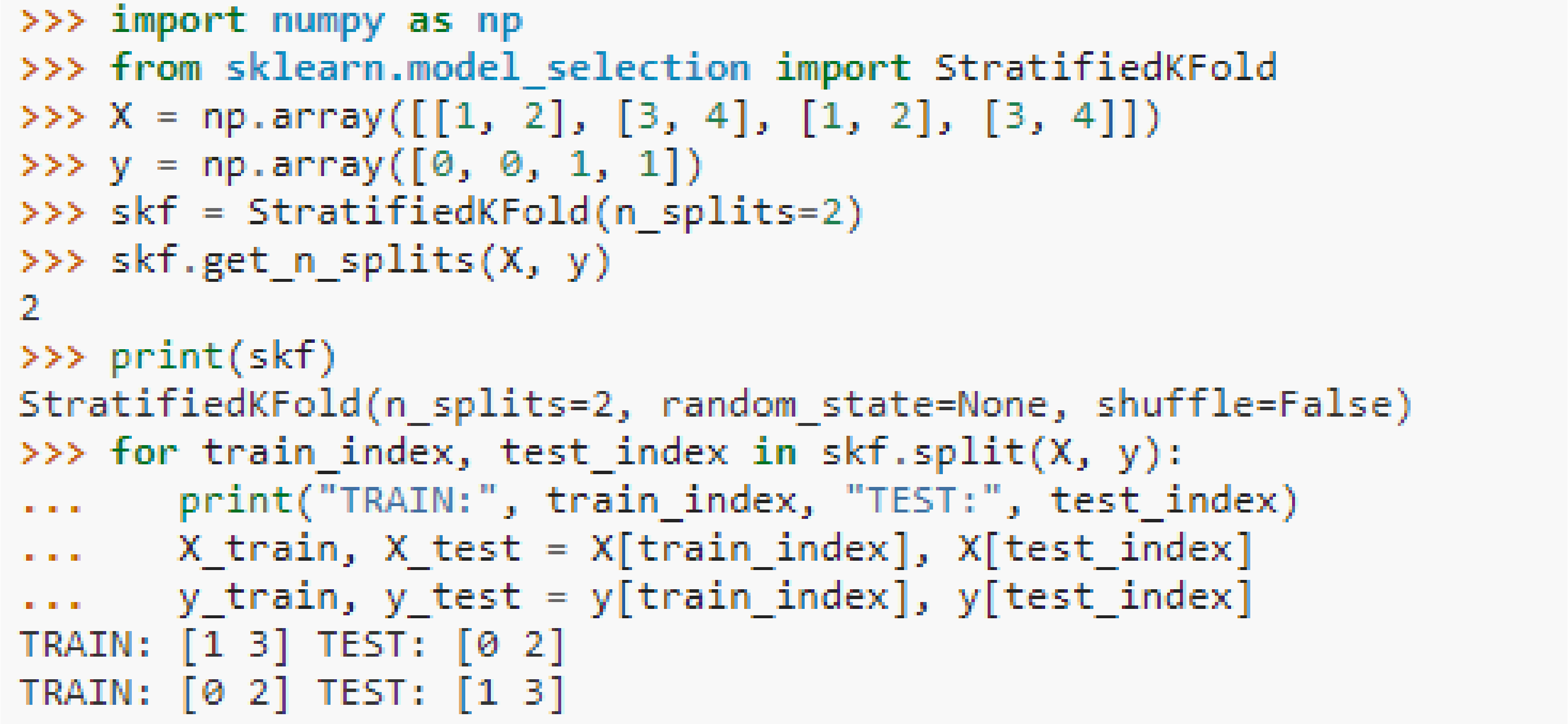
If int, random_state is the seed used by the random number generator; If RandomState instance, random_state is the random number generator; If None, the random number generator is the RandomState instance used by `np.random`. Used when `shuffle` == True.

See also:

`RepeatedStratifiedKFold`

Repeats Stratified K-Fold n times.

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html



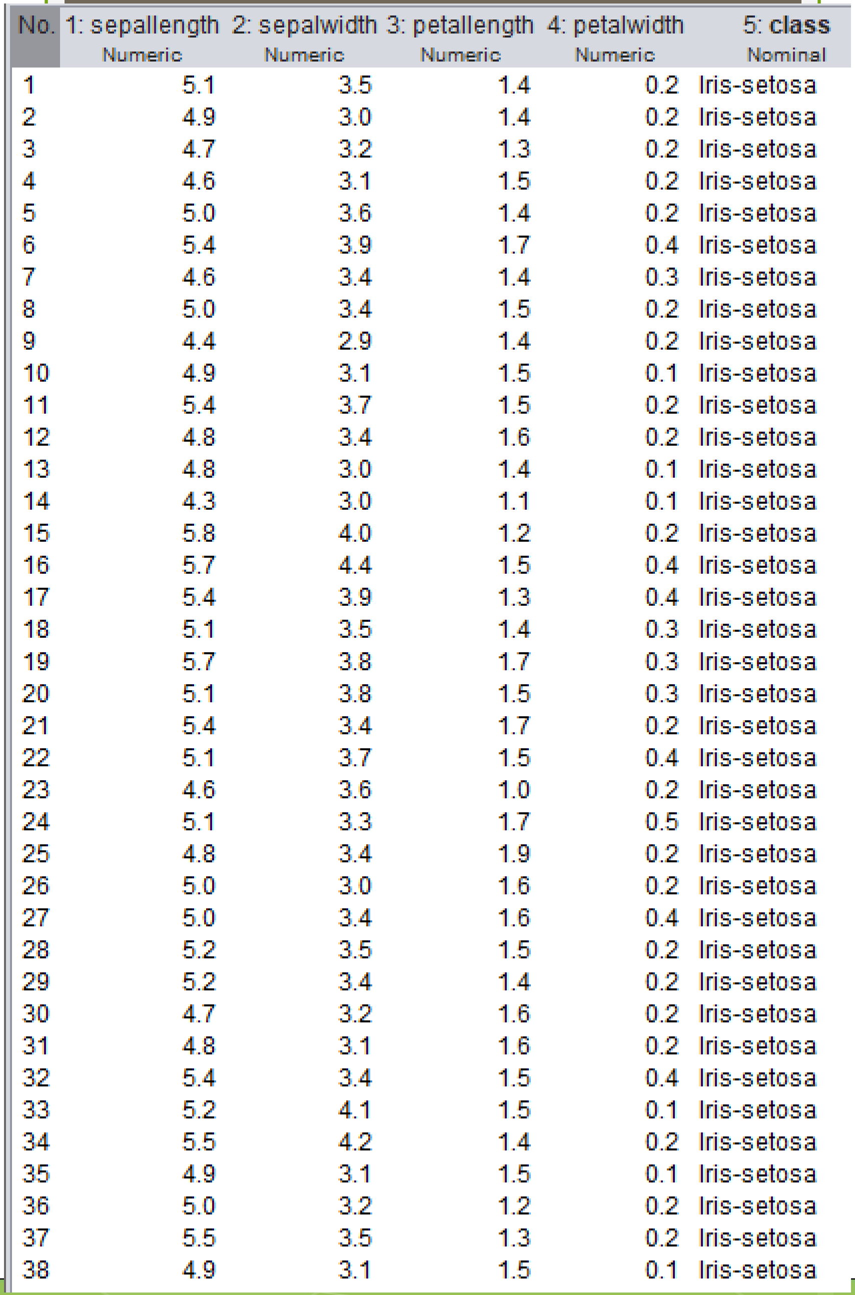
```
>>> import numpy as np
>>> from sklearn.model_selection import StratifiedKFold
>>> X = np.array([[1, 2], [3, 4], [1, 2], [3, 4]])
>>> y = np.array([0, 0, 1, 1])
>>> skf = StratifiedKFold(n_splits=2)
>>> skf.get_n_splits(X, y)
2
>>> print(skf)
StratifiedKFold(n_splits=2, random_state=None, shuffle=False)
>>> for train_index, test_index in skf.split(X, y):
...     print("TRAIN:", train_index, "TEST:", test_index)
...     X_train, X_test = X[train_index], X[test_index]
...     y_train, y_test = y[train_index], y[test_index]
TRAIN: [1 3] TEST: [0 2]
TRAIN: [0 2] TEST: [1 3]
```

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

Example – Iris Dataset

150 instances

Attributes:
sepallength: sepal length in cm
sepalwidth: sepal width in cm
petallength: petal length in cm
peatalwidth: petal width in cm



No.	1: sepallength	2: sepalwidth	3: petallength	4: petalwidth	5: class
	Numeric	Numeric	Numeric	Numeric	Nominal
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa
15	5.8	4.0	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa
21	5.4	3.4	1.7	0.2	Iris-setosa
22	5.1	3.7	1.5	0.4	Iris-setosa
23	4.6	3.6	1.0	0.2	Iris-setosa
24	5.1	3.3	1.7	0.5	Iris-setosa
25	4.8	3.4	1.9	0.2	Iris-setosa
26	5.0	3.0	1.6	0.2	Iris-setosa
27	5.0	3.4	1.6	0.4	Iris-setosa
28	5.2	3.5	1.5	0.2	Iris-setosa
29	5.2	3.4	1.4	0.2	Iris-setosa
30	4.7	3.2	1.6	0.2	Iris-setosa
31	4.8	3.1	1.6	0.2	Iris-setosa
32	5.4	3.4	1.5	0.4	Iris-setosa
33	5.2	4.1	1.5	0.1	Iris-setosa
34	5.5	4.2	1.4	0.2	Iris-setosa
35	4.9	3.1	1.5	0.1	Iris-setosa
36	5.0	3.2	1.2	0.2	Iris-setosa
37	5.5	3.5	1.3	0.2	Iris-setosa
38	4.9	3.1	1.5	0.1	Iris-setosa

Holdout with Iris Dataset

Predict the error estimation for the iris dataset using Zero R and 10 stratified folds.

Predict the error estimation for the iris dataset using Zero R and 6 stratified folds.

Holdout with Iris Dataset

Predict the error estimation for the iris dataset using Zero R and 10 stratified folds.

- Equal number of each class, so randomly selects one class, only gets 33.33% correct, 66.67% wrong.

Predict the error estimation for the iris dataset using Zero R and 6 stratified folds.

- Each fold is 25 instances , split 8,8,9. Thus Zero R will choose the type with 9. Error rate will go down.
- Weka got 32%

Leave One out Cross-Validation

Leave One Out Cross-Validation –n-fold cross-validation where n is the number of instances in the dataset

Predict the error estimation for the iris dataset using Zero R and leave one out cross-validation

Leave One out Cross-Validation

Leave One Out Cross-Validation –n-fold cross-validation where n is the number of instances in the dataset

Predict the error estimation for the iris dataset using Zero R and leave one out cross-validation

0%

Bootstrapping

Bootstrap – uses sampling with replacement to form the training set

Sample a dataset of n instances n times with replacement to form a new dataset of n instances and make this the training set

Commonly called 0.632 bootstrap because the training data will contain approximately 63.2% of the instances

Bootstrapping

Some instances are likely to be repeated in the training set.

- Given n instances, likelihood of an element being chosen to be in the training set? 1/n
- Likelihood of an element not being chosen to be in the training set? (n-1)/n or (1 – 1/n)

Pr(chosen) = 1/n

Pr(~chosen) = (n-1)/n

Bootstrapping

(n-1)/n probability of being chosen the first time.

$[(n-1)/n]^2$ probability of being chosen the first or the second time

$[(n-1)/n]^3$ probability of not being chosen the first, second or third time

.

.

.

$[(n-1)/n]^n$ probability of not being chosen in any of the n times

Bootstrapping

Repeat this process n times – likelihood of not being chosen?

$$(1 – 1/n)^n$$

$(1/2)^2 = 0.25$
$(2/3)^3 = 0.296$
$(3/4)^4 = 0.316$
$(4/5)^5 = 0.326$
$(5/6)^6 = 0.335$
$(6/7)^7 = 0.340$
$(7/8)^8 = 0.343$
$(8/9)^9 = 0.346$
$(9/10)^{10} = 0.347$

.

.

.

$(499/500)^{500} = 0.368$
$((n-1)/n)^n$ converges to 0.368

Confidence Levels

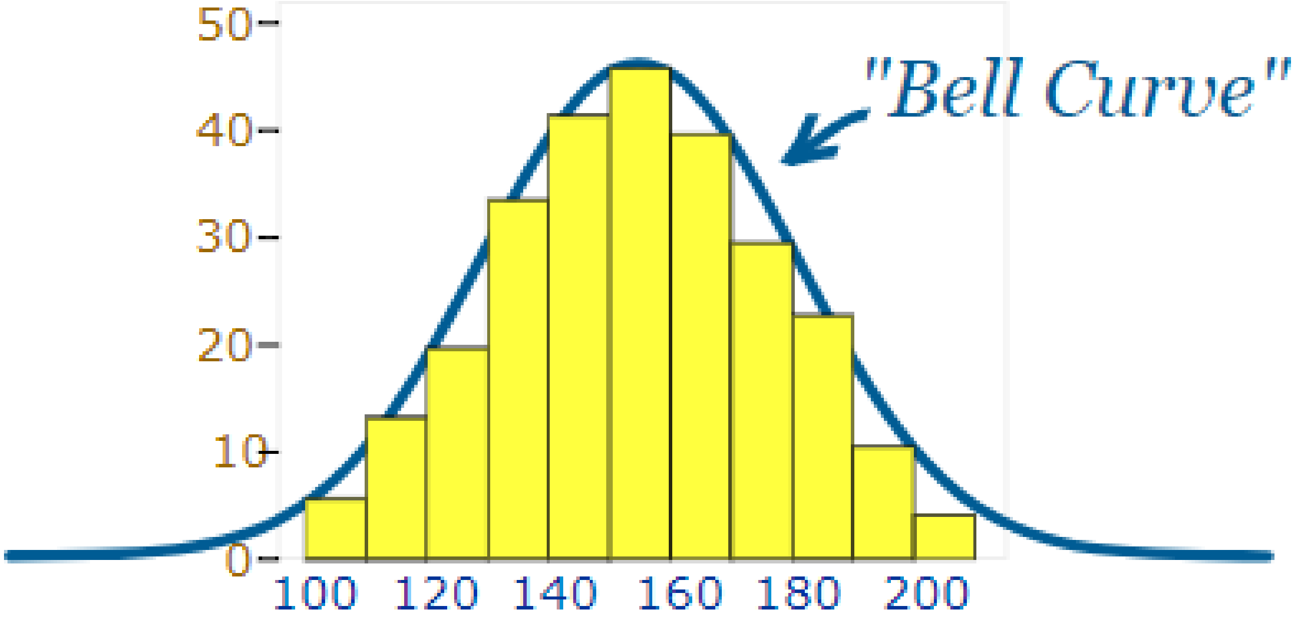
If you toss a coin 10 times and get 6 heads

Coin may or may not be biased

If you toss in 1,000 times and get 600 heads

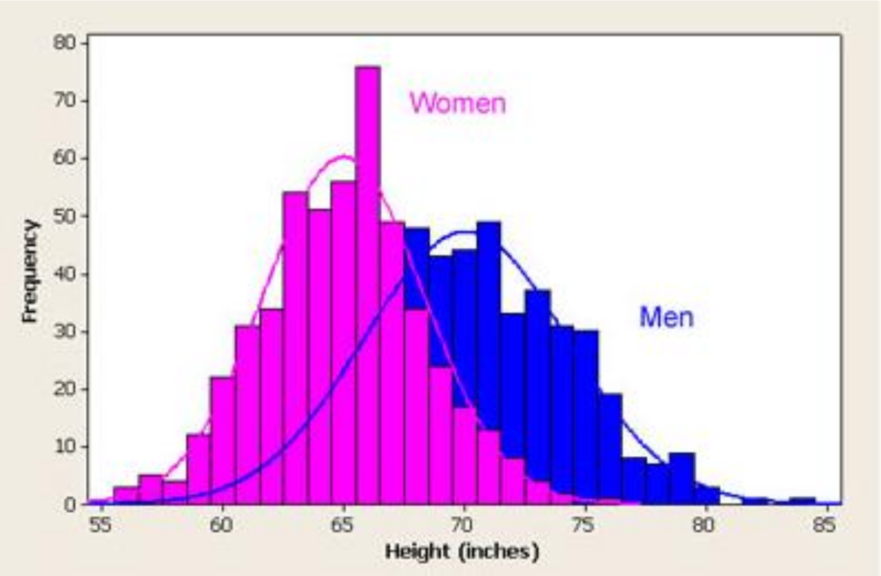
Coin is biased

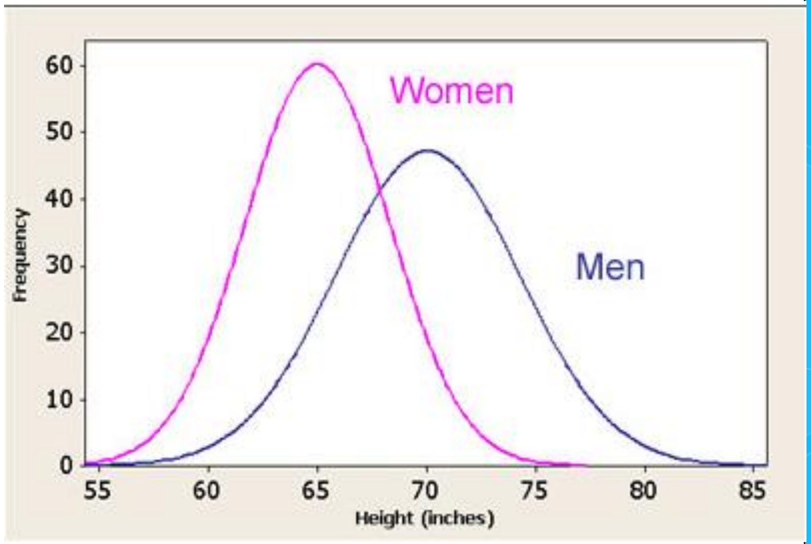
Normal Distribution



A Normal Distribution

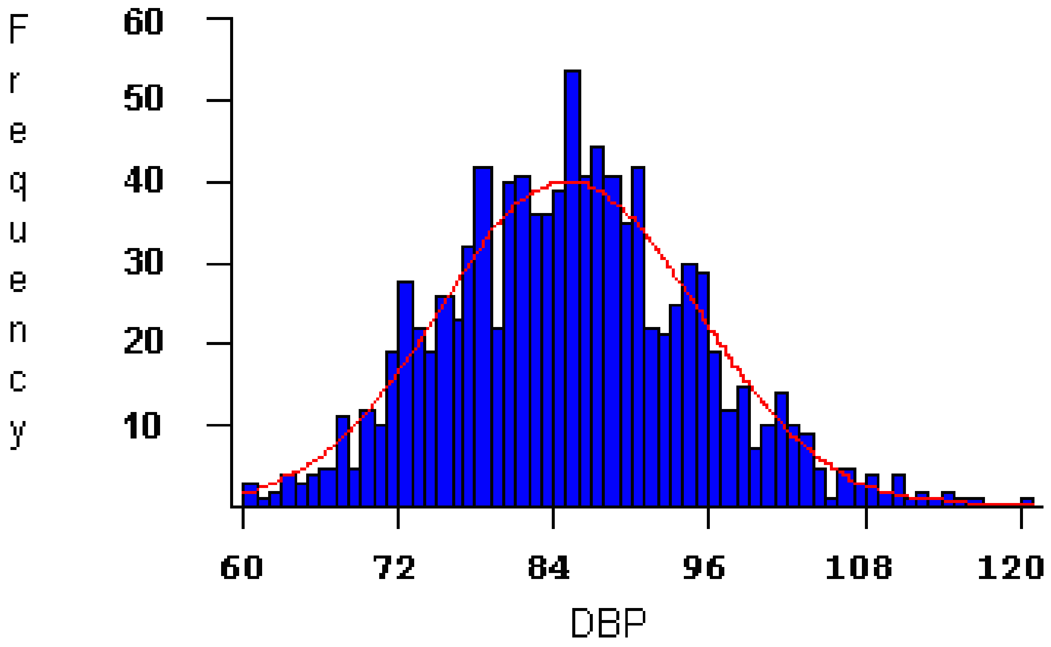
Height seen as a Normal Distribution





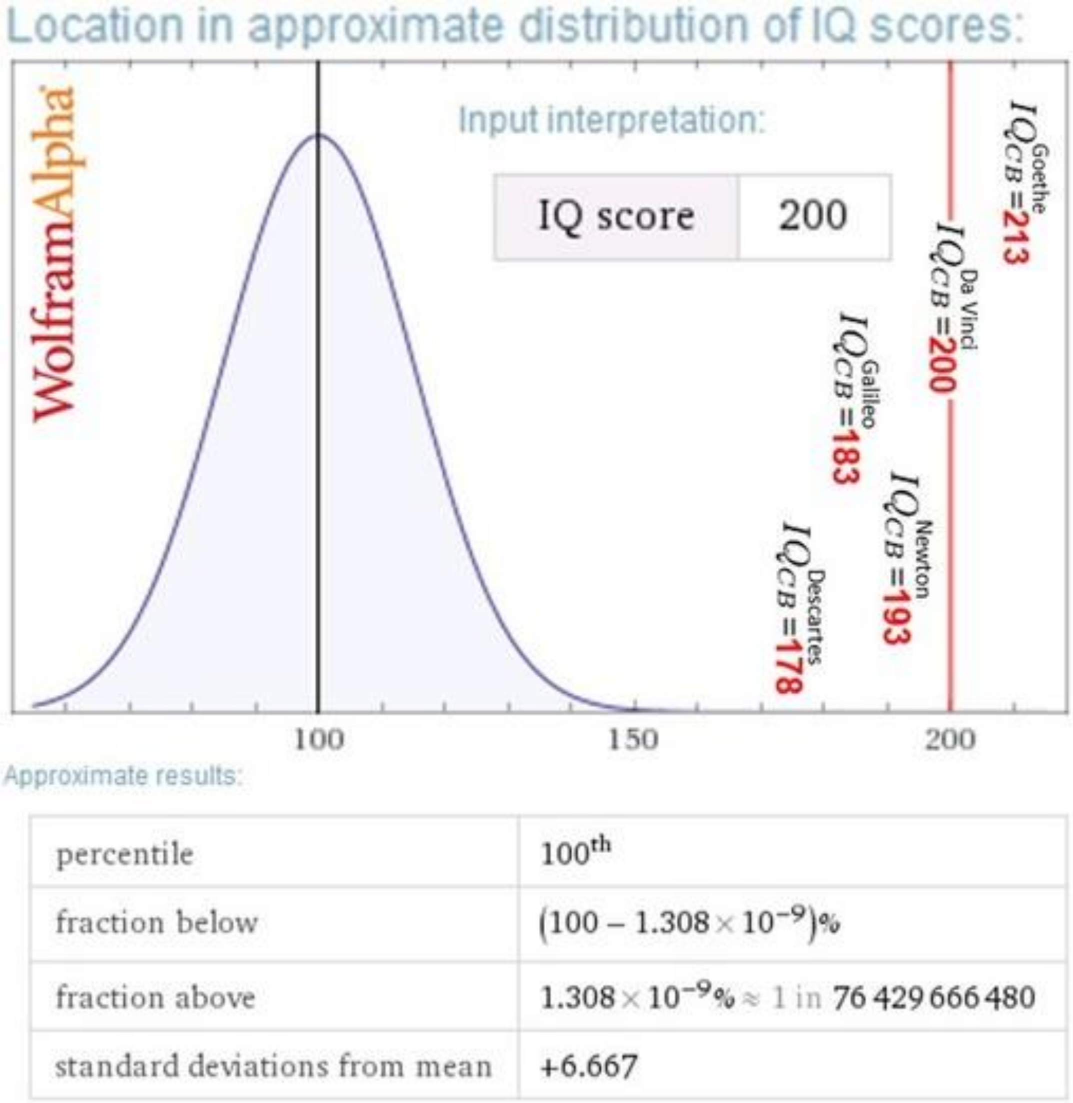
Adult male heights are on average 70 inches (5’10’”) with a standard deviation of 4 inches. Adult women are on average a bit shorter and less variable in height with a mean height of 65 inches (5’5”) and standard deviation of 3.5 inches.

Blood Pressure seen as a Normal Distribution



DBP- Diastolic Blood Pressure
Mean – 85mm
Standard deviation – 22 mm

IQ Scores - Normal Distribution



Location in approximate distribution of IQ scores:

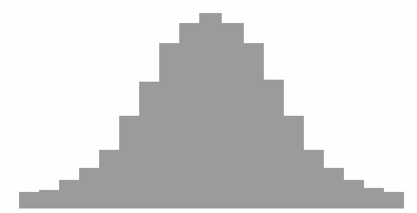
Input interpretation:

IQ score	200

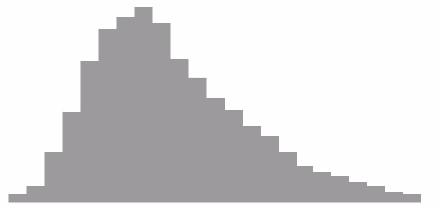
Approximate results:

percentile	100th
fraction below	$(100 - 1.308 \times 10^{-9})\%$
fraction above	$1.308 \times 10^{-9}\% \approx$ 1 in 76 429 666 480
standard deviations from mean	+6.667

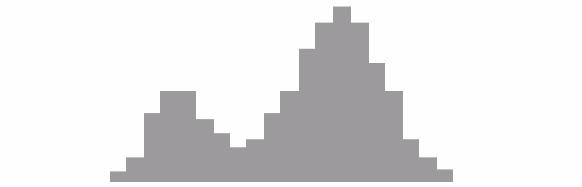
Other Distributions



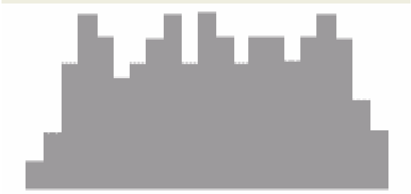
Normal distribution



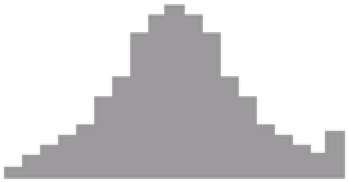
Right-skewed distribution



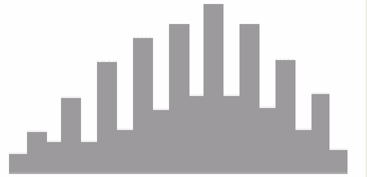
Bimodal (double-peaked) distribution



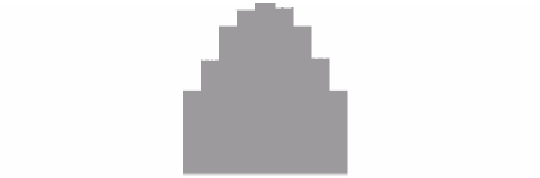
Plateau distribution



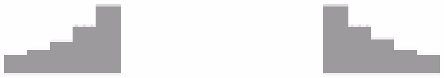
Edge peak distribution



Comb distribution

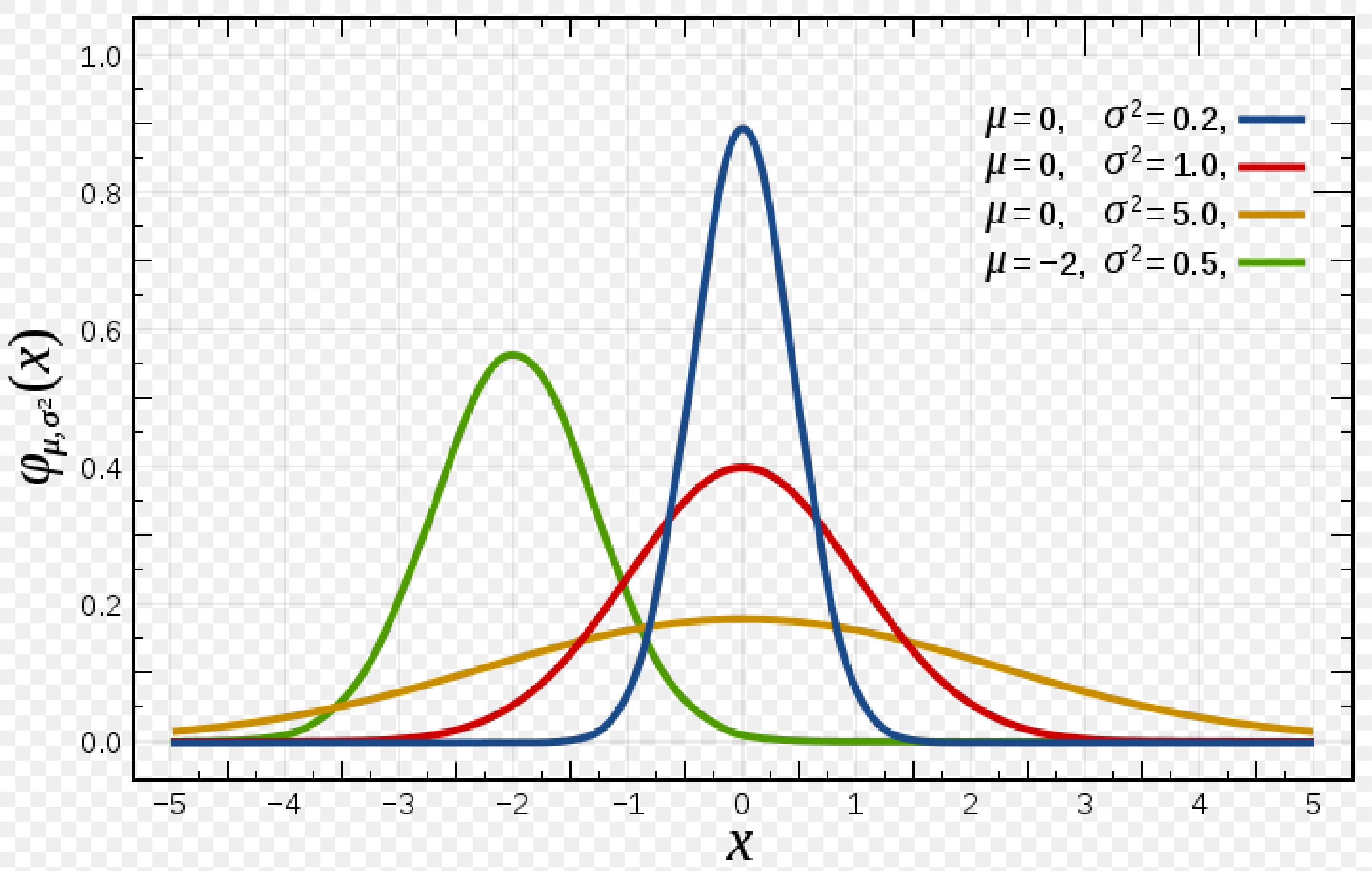


Truncated or heart-cut distribution



Dog food distribution

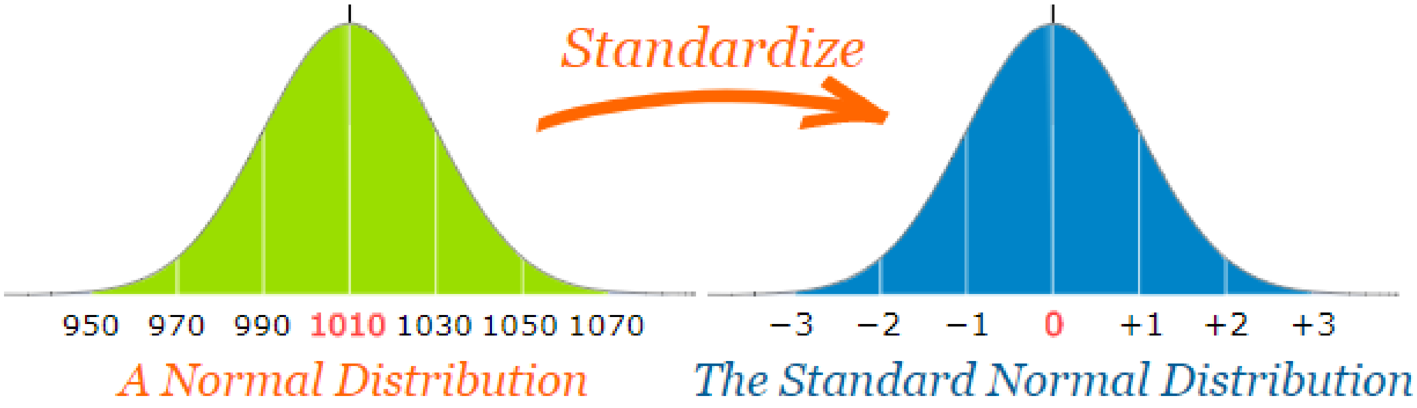
Family of Bell-Shaped Distributions



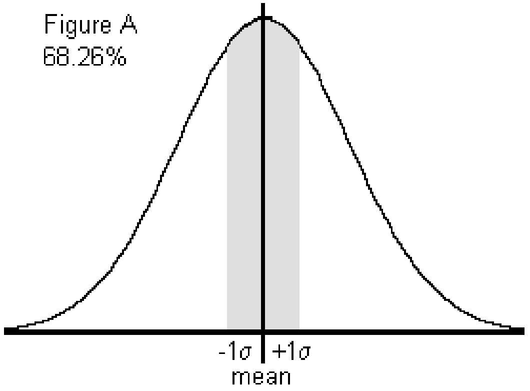
Standardized Normal Distribution

Any normalized distribution can be standardized:

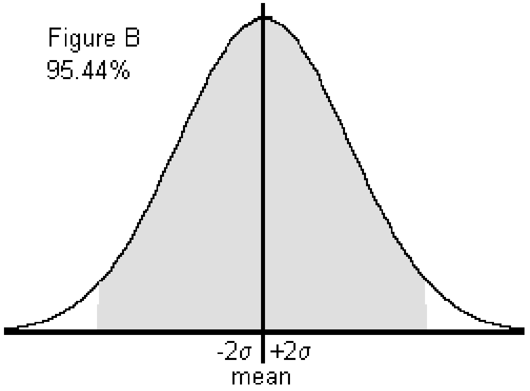
1. Subtract the mean
2. Divide by the standard deviation



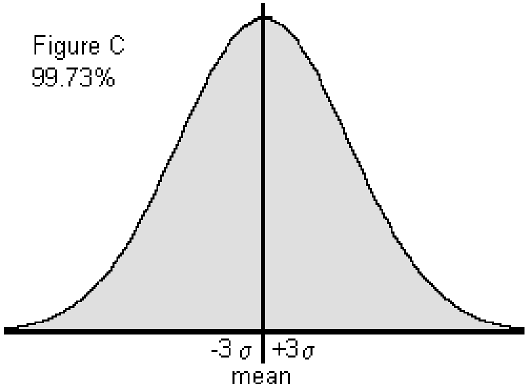
Standard Deviations



68% of all measurements fall within one standard deviation of either side of the mean



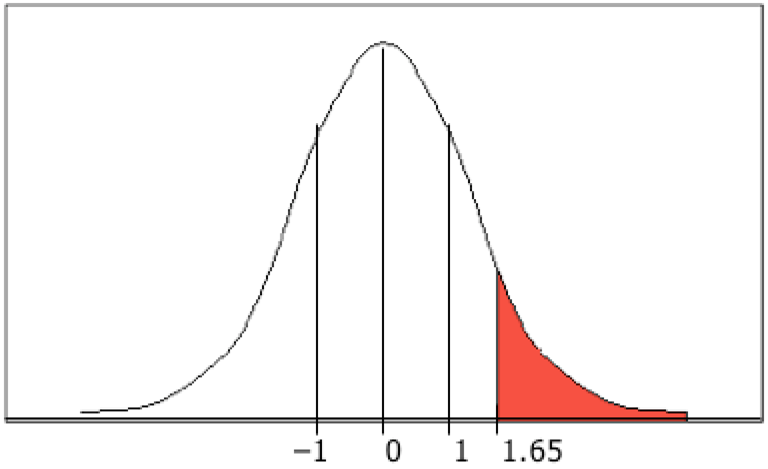
95% fall within 2 standard deviations



99.7% fall within three standard deviations

Confidence Limits

Confidence limits for the normal distribution with 0 mean and a variance of 1:



$$Pr[-1.65 \le X \le 1.65] = 90\%$$

Pr[$X \ge z$]	z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84
40%	0.25

To use this we have to reduce our random variable f to have 0 mean and unit variance

Confidence Level Example

The SAT scores from a random sample of 90 high school seniors were analyzed and found to have a mean of 545 and a standard deviation of 75. Assuming that high school senior SAT scores fall in a normal distribution, find a 90% confidence interval. In other words, tell the interval within which you are 90% confident that all scores will fall.

Confidence Level Example

Want 90% confidence so look in table Pr[X≥z] for (100%-90%)/2 = 5% which gives the z value of 1.65.

Interval is μ +/- 1.65 * (σ/sqrt(N))
= 545 +/- 1.65 * (75/sqrt(90))
= 545 +/- 13.04

Gives the interval [531.96, 558.04]. Therefore, given a score, it is 90% likely to be within the range [531.96, 558.04].