BLG 454E Machine Learning

FALL 2024-2025

Assoc. Prof. Yusuf

Design and Analysis of Machine Learning Algorithms

Introduction

- Questions:
  - Assessment of the expected error of a learning algorithm:

    Is the error rate of 1-NN less than 2%?
  - Comparing the expected errors of two algorithms:

    Is *k*-NN more accurate than MLP ?
- Training/validation/test sets

  Training errors cannot be used to compare two algorithms

  we need a validation set that is different from the training set.

Introduction

- Training/validation/test sets

  Even over a validation set though, just one run may not be enough
- The training and validation sets may be small and may contain exceptional instances
- The learning method may depend on other random factors affecting generalization
- If we do the training once, we have one learner and one validation error. To average over randomness (in training data, initial weights, etc.), we use the same algorithm and generate multiple learners.
- Resampling methods: *K*-fold cross-validation

Important Notes

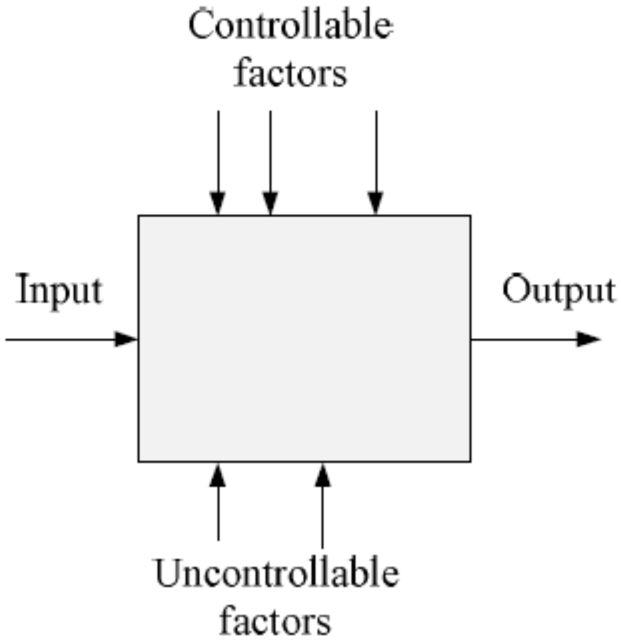
- Using our experiments we only show that a particular algorithm is better than others for this specific dataset. No algorithm can be the best on all possible datasets (see NFL (No Free Lunch) Theorems, Wolpert 1995)
- Once you decide on learning algorithm, parameter setting using the training-validation partitioned data, use ALL (training+validation) data to train your final model.
- Use a separate test set (not used for validation) to report the expected test error, not the validation error. (In papers, people do report validation error though.)

Algorithm Preference

- Criteria (Application-dependent):
    - Misclassification error, or risk (loss functions)
    - Training time/space complexity
    - Testing time/space complexity
    - Interpretability
    - Easy programmability
- Cost-sensitive learning: Takes other costs into account.

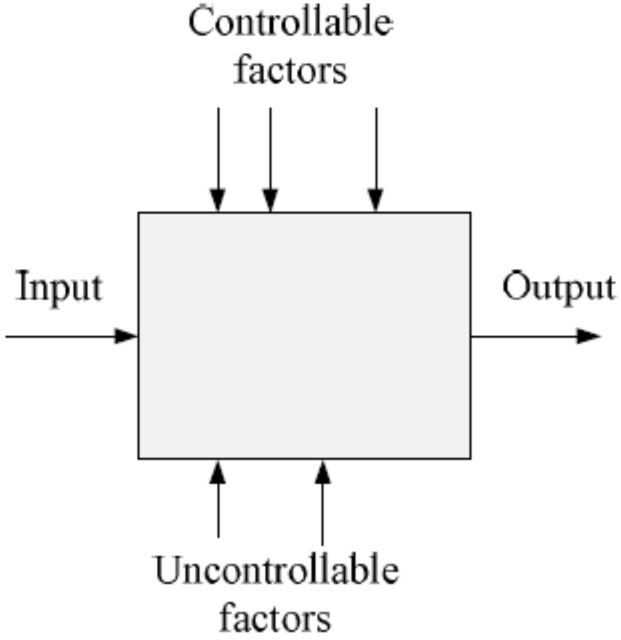
Factors and Response

- A learner is trained on a dataset and generates an output for a given input.
- An experiment is a test or a series of tests where we play with the factors that affect the output.
- Factors may be the algorithm used, the training set, input features, and so on, and we observe the changes in the response to be able to extract information
- Our aim is to plan and conduct machine learning experiments and analyze the data resulting from the experiments, to be able to eliminate the effect of chance and obtain conclusions which we can consider statistically significant.



Factors and Response

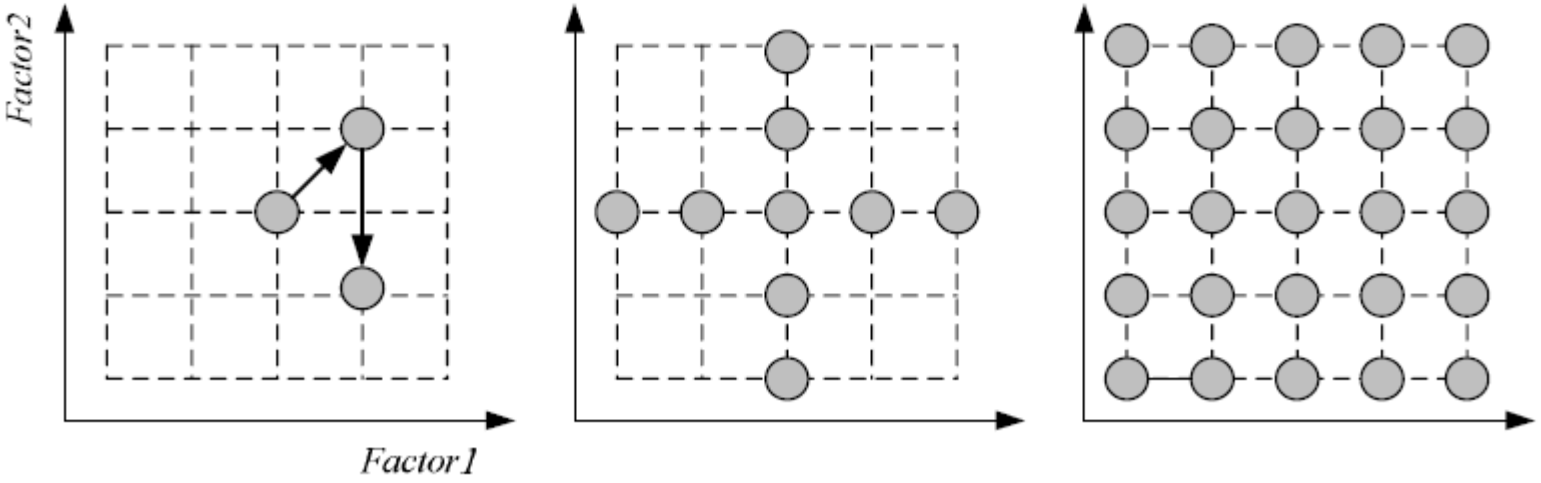
- The controllable factors, as the name suggests, are those we have control on.
- There are also the hyperparameters of the algorithm, for example, the number of hidden units for a multilayer perceptron, k for k-nearest neighbor, C for support vector machines, and so on.
- Uncontrollable factors: these are the noise in the data, the particular training subset if we are resampling from a large set, randomness in the optimization process, for example, the initial state in gradient descent with multilayer perceptrons, and so on.



Factors and Response: Example

- We may be using principal components analyzer (PCA) to reduce dimensionality to d before a k-nearest neighbor (k-NN) classifier
- Controllable Factors:
  - PCA to reduce dimension to d
  - Knn classifier with k
- Response:
  - Classification error on validation set
- Find the setting of k and d for the best response

Strategies of Experimentation



(a) Best guess (b) One factor at a time (c) Factorial design

a) we start at some setting of the factors that we
believe is a good configuration

b) it assumes that there is no *interaction* between the factors, which may not always be true.

Grid search
Best approach
F Factors with L levels each:
Cost: $O(L^F)$ (☹ !!!)

Response surface design for approximating and maximizing the response function in terms of the controllable factors

Randomization, Replication, and Blocking

- Randomization requires that the order in which the runs are carried out should be randomly determined so that the results are independent. Ordering generally is not a problem in software experiments.
- Replication implies that for the same configuration of (controllable) factors, the experiment should be run a number of times to average over the effect of uncontrollable factors. In machine learning, this is typically done by running the same algorithm on a number of resampled versions of the same dataset; this is known as cross-validation,

Randomization, Replication, and Blocking

- *Blocking* is used to reduce or eliminate the variability due to *nuisance factors* that influence the response but in which we are not interested.
- In machine learning experimentation, when we use resampling and use different subsets of the data for different replicates, we need to make sure that for example if we are comparing learning algorithms, they should all use the same set of resampled subsets.

Guidelines for ML experiments

The steps in machine learning are the same as for any type of experimentation

A. Aim of the study: we may be interested in assessing the expected error of a learning algorithm on a particular problem. Given two learning algorithms and a particular problem as defined by a dataset, we may want to determine which one has less generalization error.

B. Selection of the response variable: We need to decide on what we should use as the quality measure. Most frequently, error is used that is the misclassification error for classification and mean square error for regression.

C. Choice of factors and levels: What the factors are depend on the aim of the study. If we fix an algorithm and want to find the best hyperparameters, then those are the factors. If we are comparing algorithms, the learning algorithm is a factor. If we have different datasets, they also become a factor

Guidelines for ML experiments

D. Choice of experimental design: It is always better to do a factorial design unless we are sure that the factors do not interact, because mostly they do. Replication number depends on the dataset size;

It is also important to avoid as much as possible toy, synthetic data and use datasets that are collected from real-world under real-life circumstances

E. Performing the experiment

In a large experiment, it is always a good idea to save intermediate results (or seeds of the random number generator), so that a part of the whole experiment can be rerun when desired.

In comparing one’s favorite algorithm with a competitor, both should be investigated equally diligently.

Guidelines for ML experiments

F. Statistical Analysis of the Data

This corresponds to analyzing data in a way so that whatever conclusion we get is not subjective or due to chance.

G. Conclusions and Recommendations

Most statistical, and hence machine learning or data mining, studies are iterative. It is for this reason that we never start with all the experimentation.

Resampling and K-Fold Cross-Validation

- If sample X is large enough we can randomly divide it into *K* parts, then randomly divide each part into two and use one half for training and the other half for validation.
- Unfortunately, datasets are never large enough to do this. So we should do our best with small datasets.
- This is done by repeated use of the same data split differently; this is called *crossvalidation*.
- We also need to make sure that classes are represented in the right proportions when subsets of data are held out, not to disturb the class prior probabilities; this is called *stratification*.

Resampling and K-Fold Cross-Validation

- The need for multiple training/validation sets
  $\{X_i, V_i\}_i$: Training/validation sets of fold *i*
- *K*-fold cross-validation: Divide X into *k*, X_i,*i*=1,...,*K*

$$
\begin{aligned}
&\mathcal{V}_1 = \mathcal{X}_1 \quad \mathcal{T}_1 = \mathcal{X}_2 \cup \mathcal{X}_3 \cup \cdots \cup \mathcal{X}_K \\
&\mathcal{V}_2 = \mathcal{X}_2 \quad \mathcal{T}_2 = \mathcal{X}_1 \cup \mathcal{X}_3 \cup \cdots \cup \mathcal{X}_K \\
&\quad\vdots \\
&\mathcal{V}_K = \mathcal{X}_K \quad \mathcal{T}_K = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \cdots \cup \mathcal{X}_{K-1}
\end{aligned}
$$

- T_i share *K*-2 parts

5×2 Cross-Validation

- 5 times 2 fold cross-validation (Dietterich, 1998)

$$\mathcal{T}_1 = \mathcal{X}_1^{(1)} \qquad \mathcal{V}_1 = \mathcal{X}_1^{(2)}$$

$$\mathcal{T}_2 = \mathcal{X}_1^{(2)} \qquad \mathcal{V}_2 = \mathcal{X}_1^{(1)}$$

$$\mathcal{T}_3 = \mathcal{X}_2^{(1)} \qquad \mathcal{V}_3 = \mathcal{X}_2^{(2)}$$

$$\mathcal{T}_4 = \mathcal{X}_2^{(2)} \qquad \mathcal{V}_4 = \mathcal{X}_2^{(1)}$$

$$\vdots$$

$$\mathcal{T}_9 = \mathcal{X}_5^{(1)} \qquad \mathcal{V}_9 = \mathcal{X}_5^{(2)}$$

$$\mathcal{T}_{10} = \mathcal{X}_5^{(2)} \qquad \mathcal{V}_{10} = \mathcal{X}_5^{(1)}$$

5×2 Cross-Validation

- After five folds, the sets share many instances and overlap so much that the statistics calculated from these sets, namely, validation error rates, become too dependent and do not add new information.
- If we do have fewer than five folds, we get less data (fewer than ten sets) and will not have a large enough sample to fit a distribution to and test our hypothesis on.

Bootstrapping

- To generate multiple samples from a single sample, an alternative to cross-validation is the *bootstrap* that generates new samples by drawing instances from the original sample *with* replacement.
- The bootstrap samples may overlap more than cross-validation samples and hence their estimates are more dependent; but is considered the best way to do resampling for very small datasets.

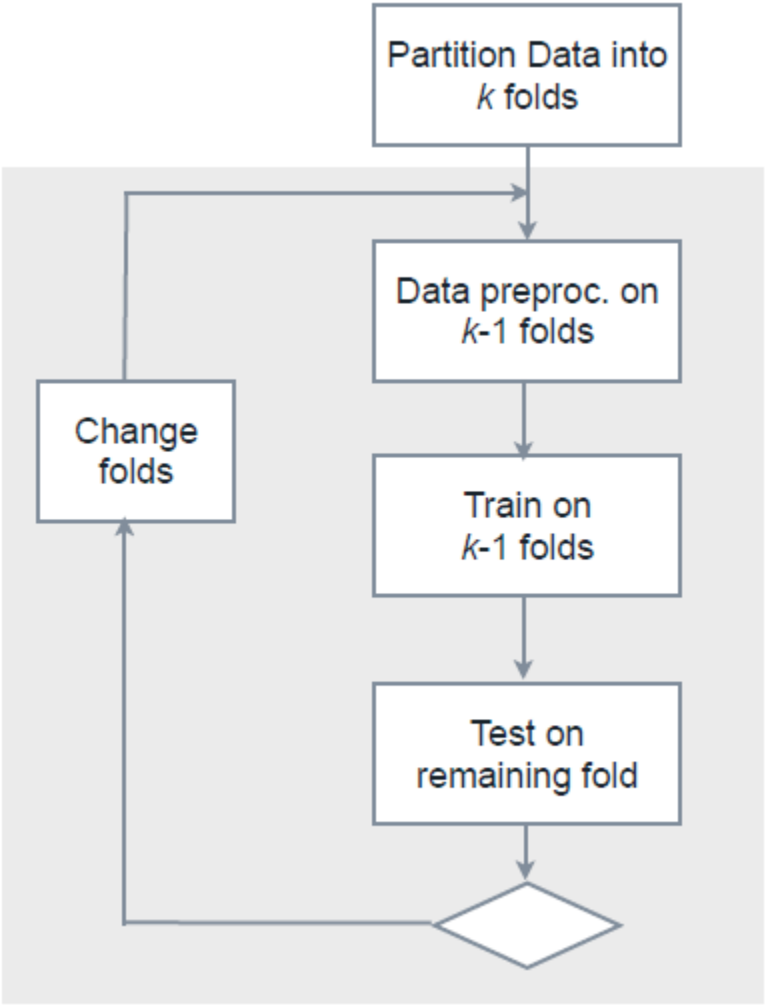
Bootstrapping

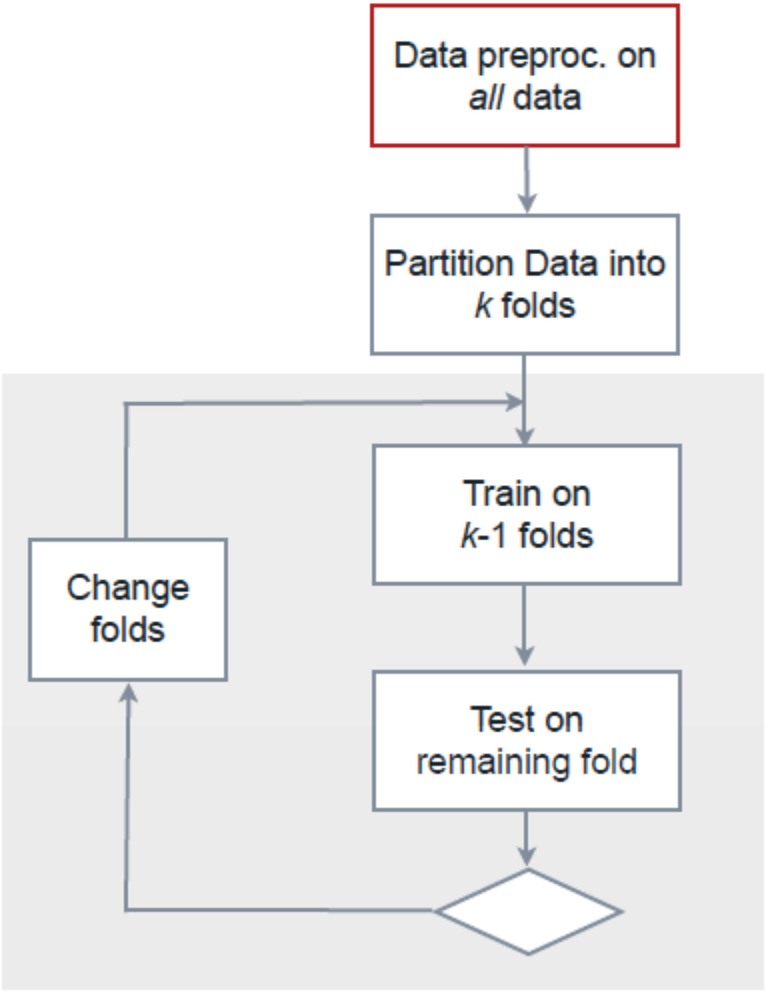
- Draw instances from a dataset *with replacement*
- Prob that we do not pick an instance after N draws

$$\left(1-\frac{1}{N}\right)^N \approx e^{-1} = 0.368$$

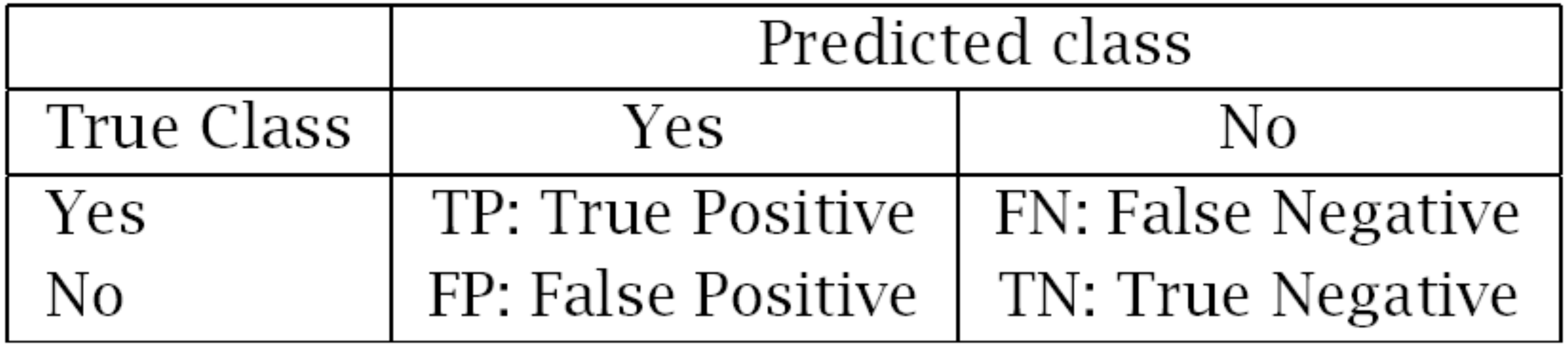
- The training data contains approximately 63.2 percent of the instances.
- The error estimate will be pessimistic. The solution is replication, that is, to repeat the process many times and look at the average behavior.

Test Data Leaks into Training





Measuring Error



	Predicted class	
True Class	Yes	No
Yes	TP: True Positive	FN: False Negative
No	FP: False Positive	TN: True Negative

- Error rate = # of errors / # of instances = (FN+FP) / N
- Recall = # of found positives / # of positives
  = TP / (TP+FN) = sensitivity = hit rate
- Precision = # of found positives / # of found
  = TP / (TP+FP)
- Specificity = TN / (TN+FP)
- False alarm rate = FP / (FP+TN) = 1 - Specificity

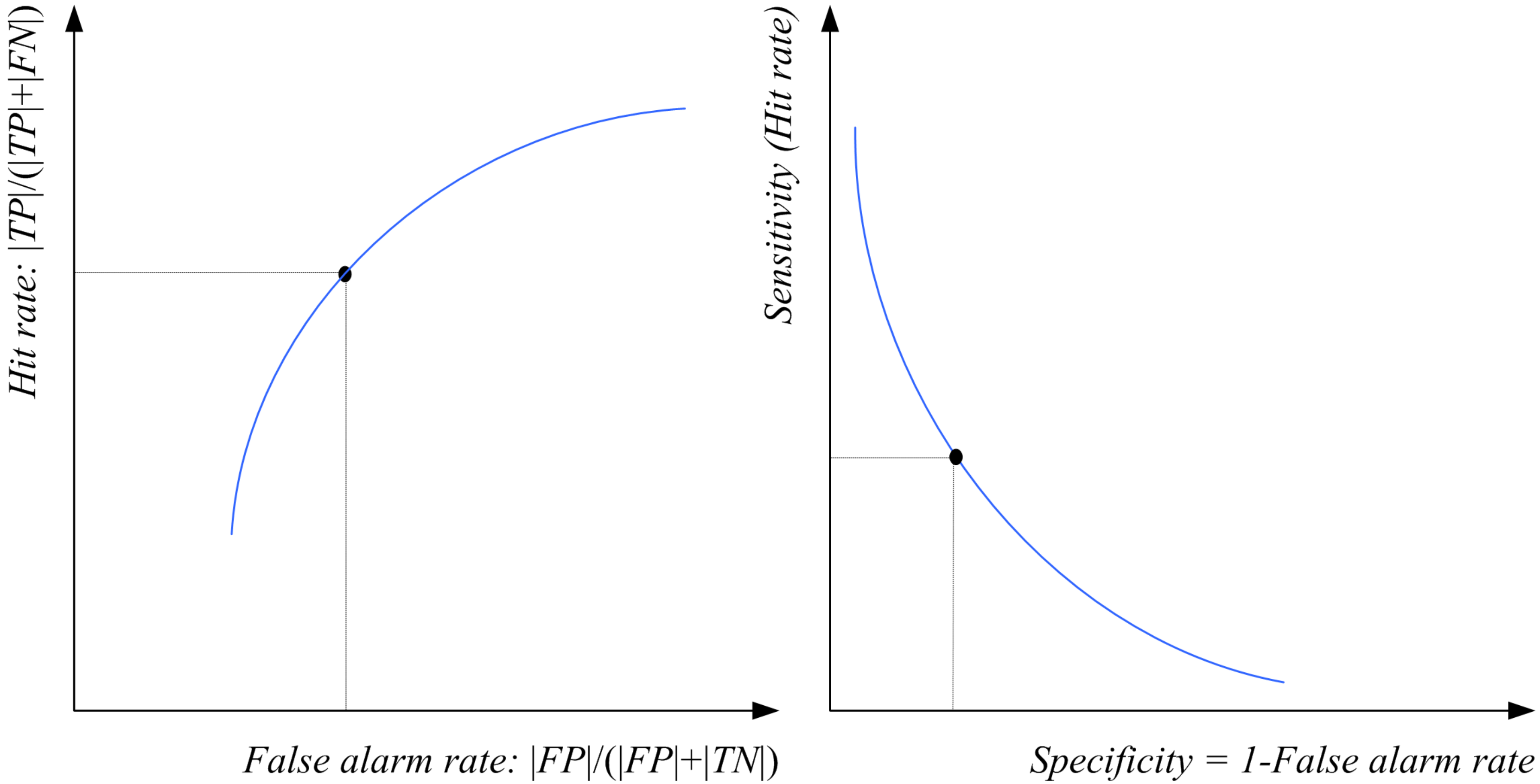
Example:

users log on to their accounts by voice.
A false positive is wrongly logging on an impostor
a false negative is refusing a valid user

Measuring Error

- True positive rate, *tp-rate*, also known as *hit rate*, measures what proportion of valid users we authenticate a
- false positive rate, *fp-rate*, also known as *false alarm rate*, is the proportion of impostors we wrongly accept
- $P(C1|x)$, the probability of the positive class, and for the negative class, we have $P(C2|x) = 1 - P(C1|x)$, and we choose "positive" if $P(C1|x) > \vartheta$.
- If ϑ is close to 1, we hardly choose the positive class; that is, we will have no false positives but also few true positives. As we decrease ϑ to increase the number of true positives, we risk introducing false positives

ROC Curve

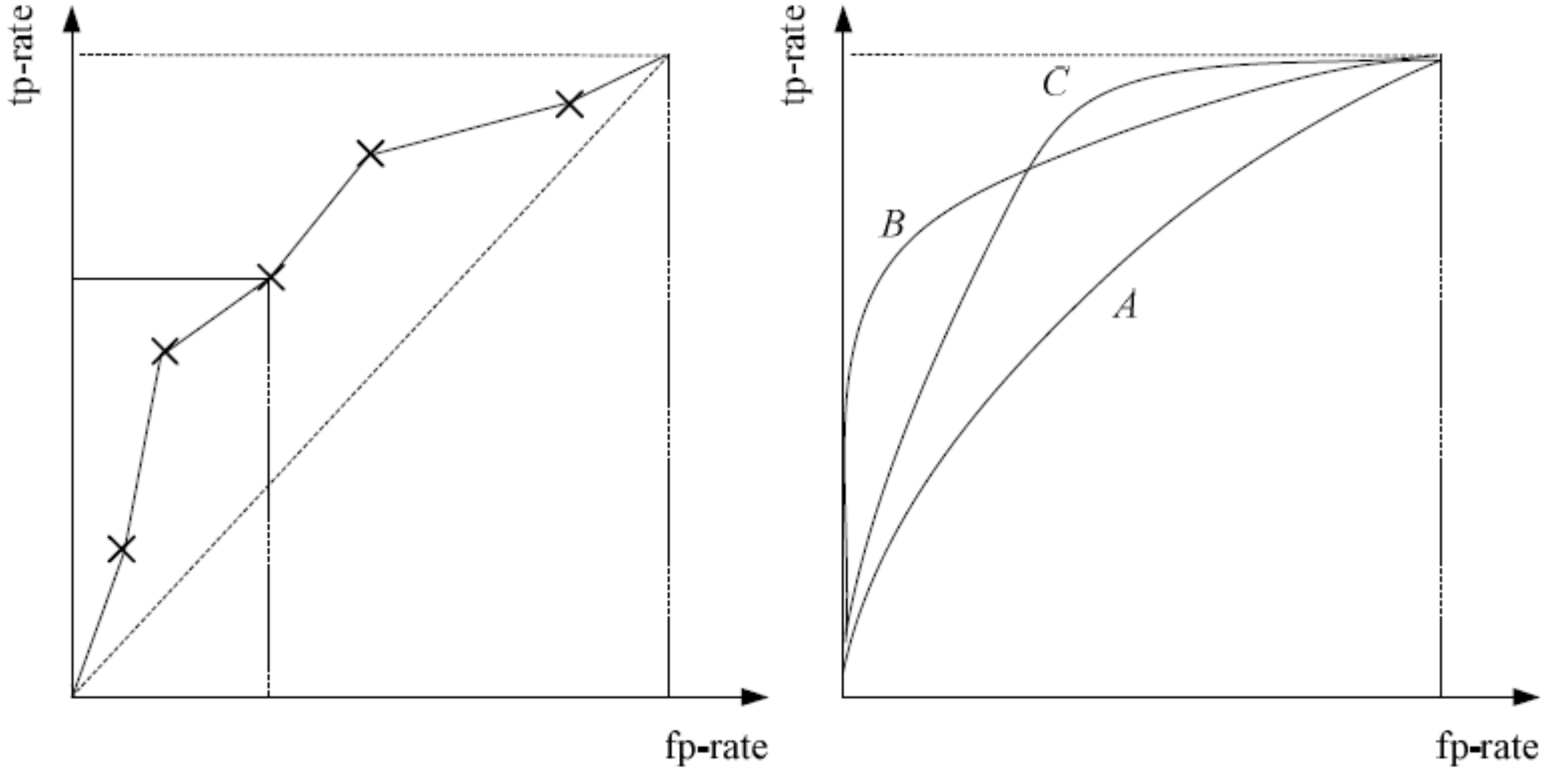

Hit rate: |TP|/(|TP|+|FN|)
False alarm rate: |FP|/(|FP|+|TN|)
Sensitivity (Hit rate)
Specificity = 1-False alarm rate

ROC Curve

- For different values of ϑ, we can get a number of pairs of (tp-rate, fp-rate) values and by connecting them we get the *receiver operating* characteristics *characteristics* (ROC) curve.
- Different values of ϑ correspond to different loss matrices for the two types of error and the ROC curve can also be seen as the behavior of a classifier under different loss matrices.
- Ideally, a classifier has a tp-rate of 1 and a fp-rate of 0, and hence a classifier is better the more it gets closer to the upper-left corner.

ROC Curve

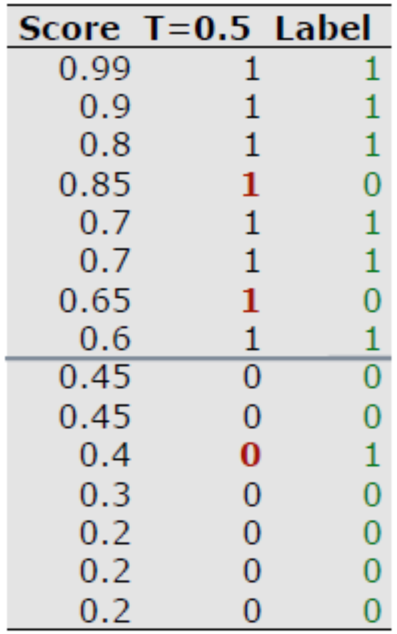
- On the diagonal, we make as many true decisions as false ones, and this is the worst one can do (any classifier that is below the diagonal can be improved by flipping its decision).
- ROC allows a visual analysis; if we want to reduce the curve to a single the number we can do this by calculating the *area under the curve* (AUC)
- A classifier ideally has an AUC of 1 and AUC values of different classifiers can be compared to give us a general performance averaged over different loss conditions.



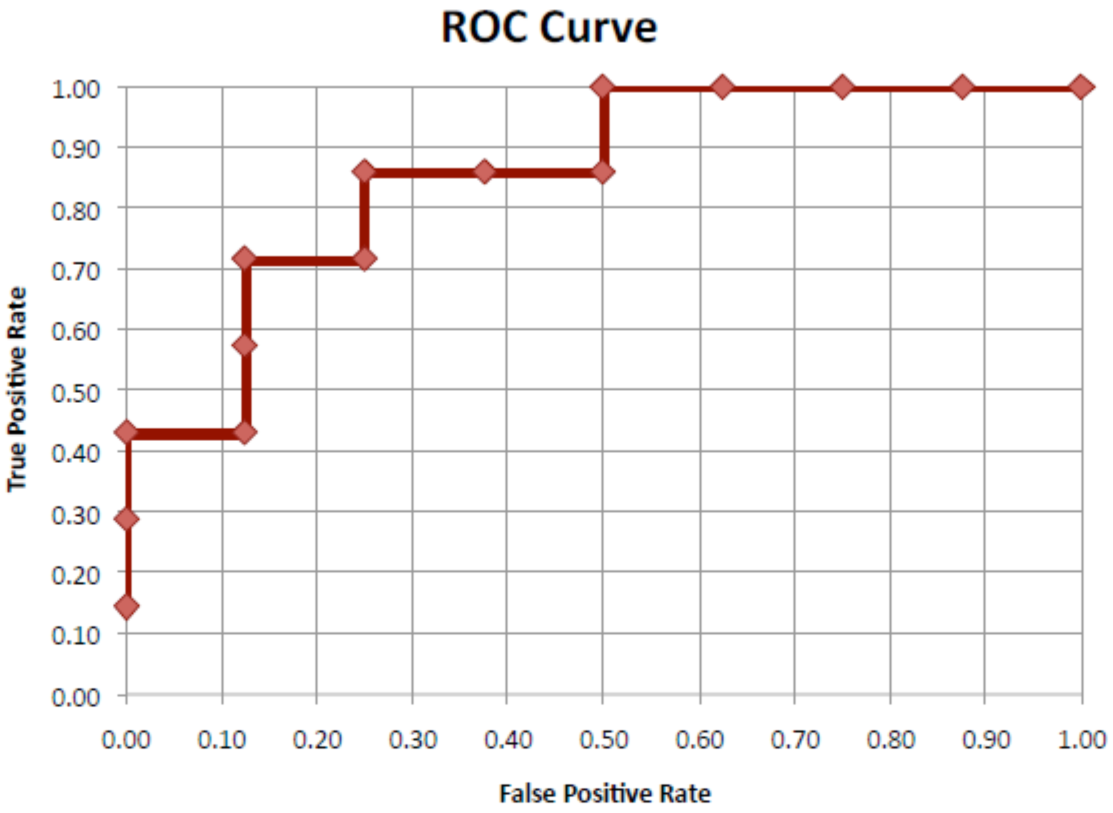
(a) Example ROC curve

(b) Different ROC curves for different classifiers

ROC Curve



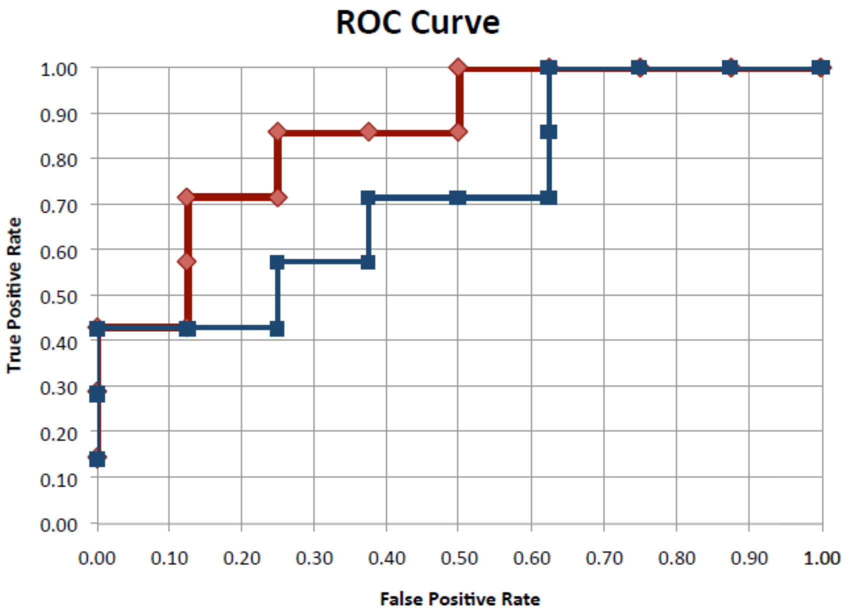
Score	T=0.5	Label
0.99	1	1
0.9	1	1
0.8	1	1
0.85	1	0
0.7	1	1
0.7	1	1
0.65	1	0
0.6	1	1
0.45	0	0
0.45	0	0
0.4	0	1
0.3	0	0
0.2	0	0
0.2	0	0
0.2	0	0

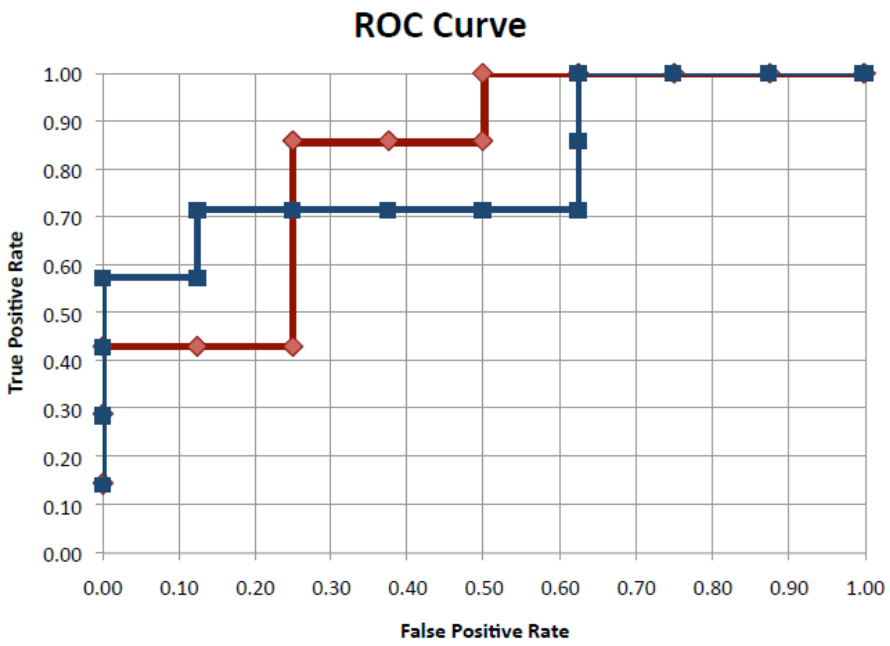


A ranking classifier will score test samples with probability of belonging to positive class

Classifier with threshold T=0.5 is only one of a family of classifiers that can be derived from this ranking

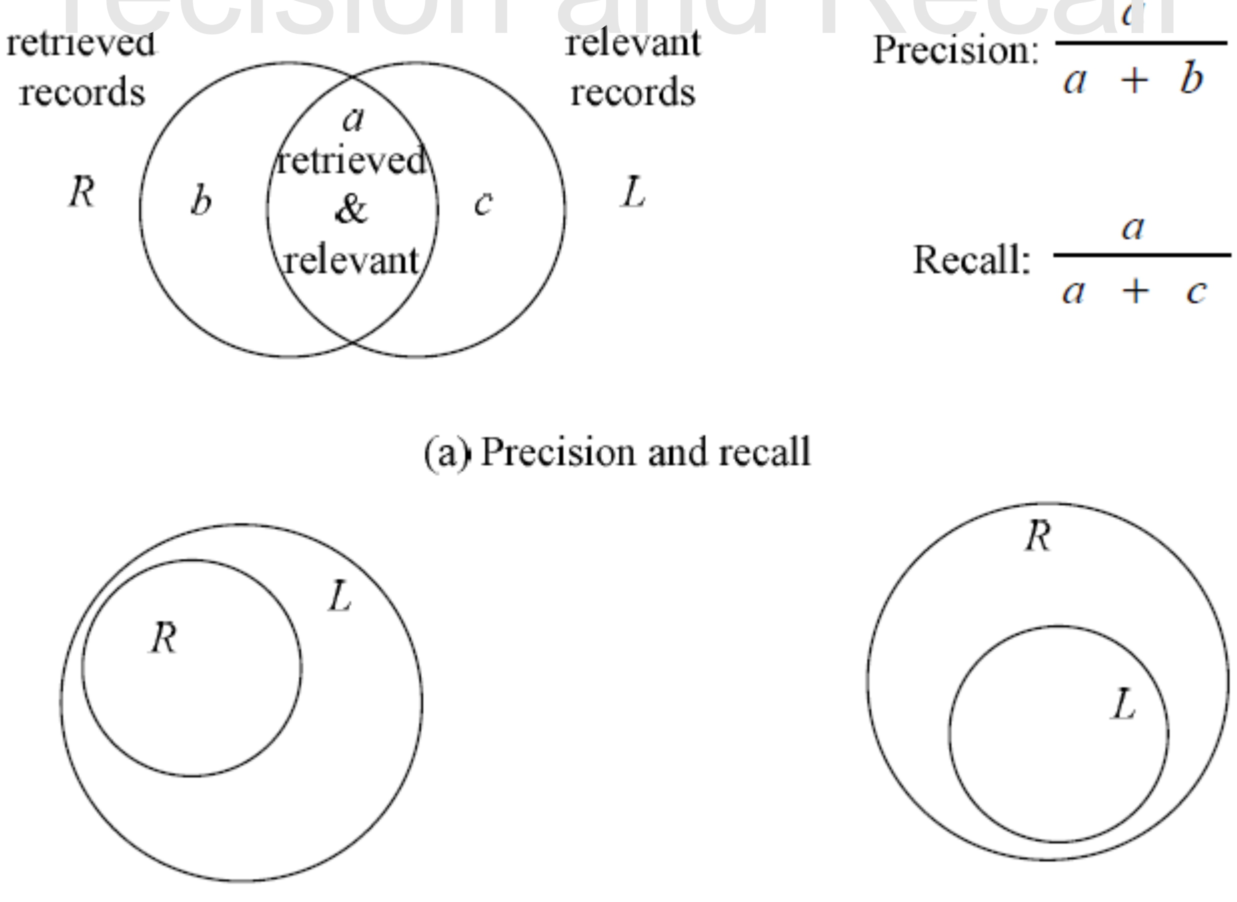
Comparing ROC Curves





- One classifier can dominate another across the range of Threshold.
- Or the situation can be less clear
- The area under the curve (AUC) can be used to keep the score

Precision and Recall



(a) Precision and recall

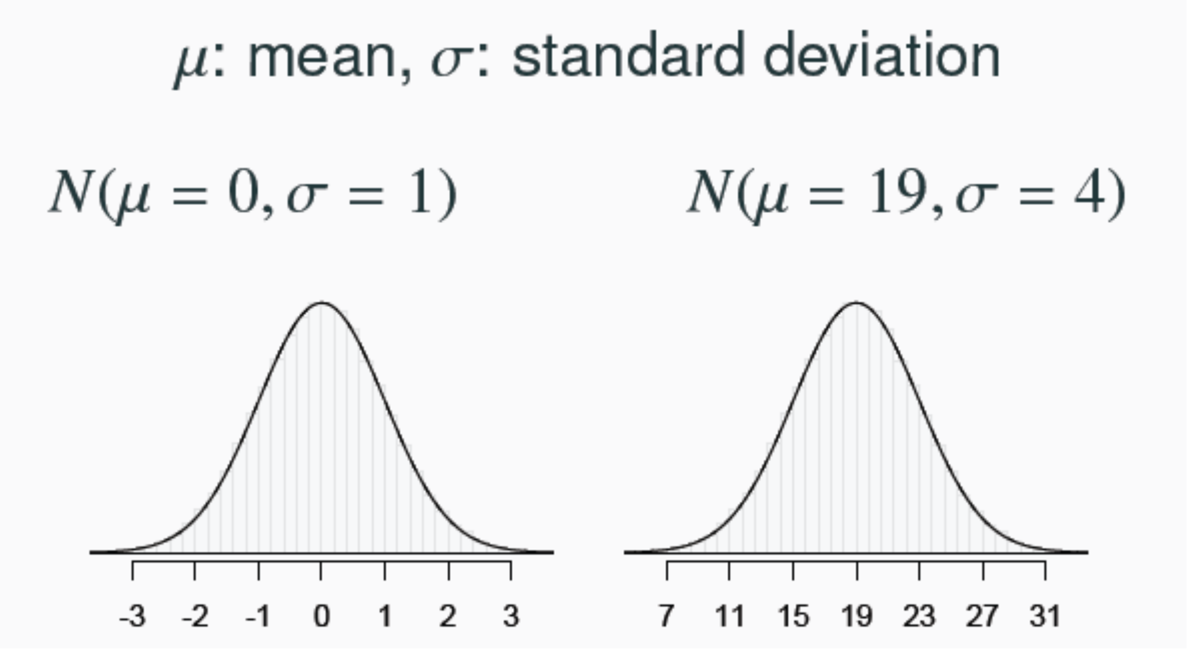
(b) Precision = 1

(c) Recall = 1

(b) Precision
is 1; all the retrieved records are relevant but there may be relevant ones
not retrieved.
(c) Recall is 1; all the relevant records are retrieved but there may
also be irrelevant records that are retrieved.

Interval Estimation

Normal distributions with different parameters



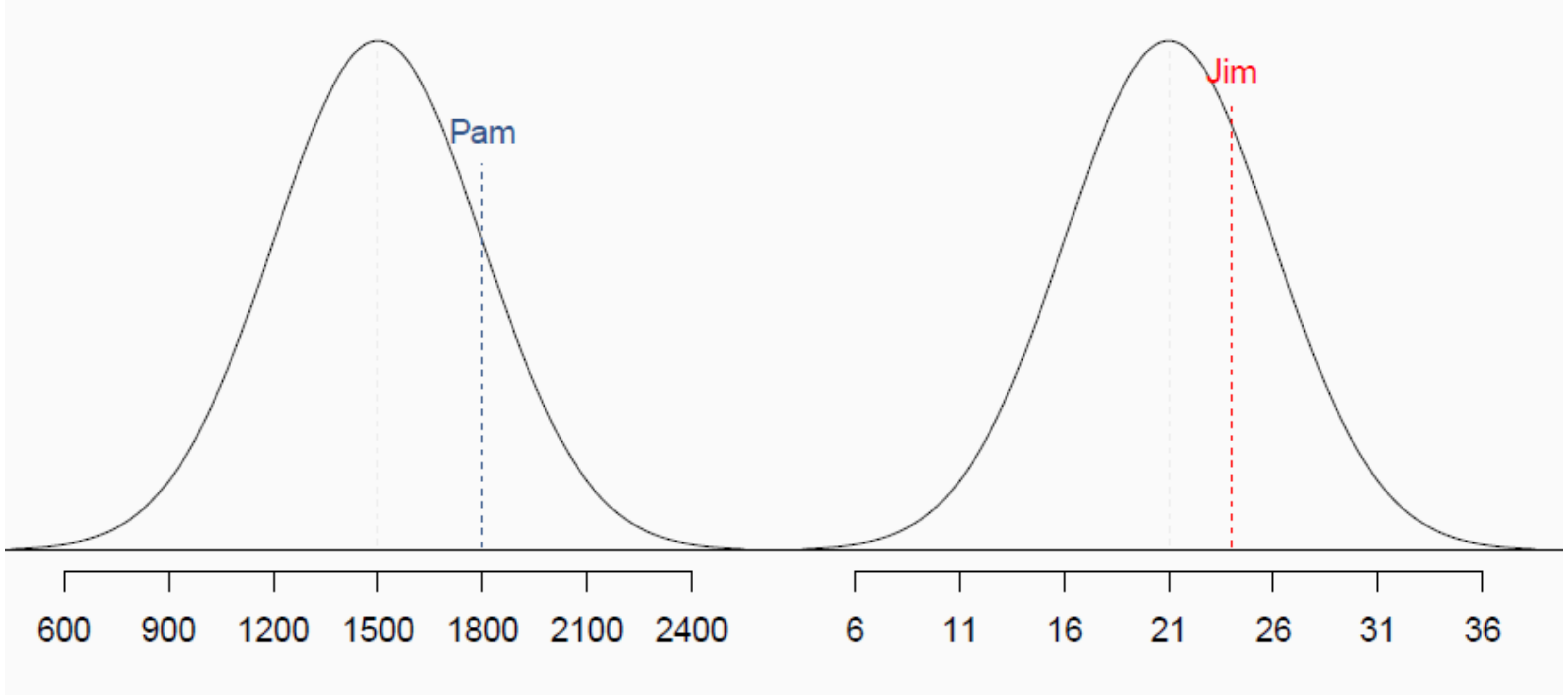
μ: mean, σ: standard deviation

$N(\mu = 0, \sigma = 1)$

$N(\mu = 19, \sigma = 4)$

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Example 1: SAT scores are distributed nearly normally with mean 1500 and standard deviation 300. ACT scores are distributed nearly normally with mean 21 and standard deviation 5. A college admissions officer wants to determine which of the two applicants scored better on their standardized test with respect to the other test takers: Pam, who earned an 1800 on her SAT, or Jim, who scored a 24 on his ACT?



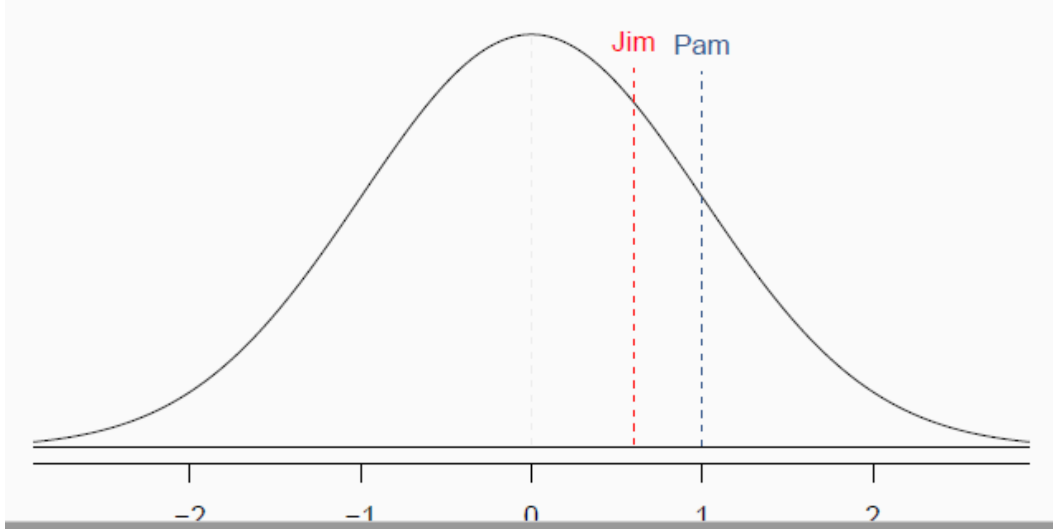
This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Standardizing with Z Scores

Since we cannot just compare these two raw scores, we instead compare how many standard deviations beyond the mean each observation is.

Pam’s score is (1800-1500)/300 = 1 standard deviation is above the mean

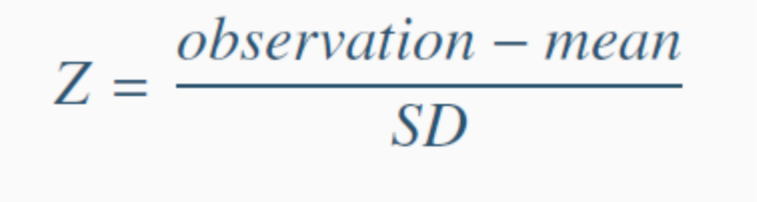
Jim’s score is (24-21)/5 = 0.6 standard deviations above the mean.



This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Standardizing with Z Scores

- These are called standardized scores, or Z scores.
- Z score of an observation is the number of standard deviations it falls above or below the mean

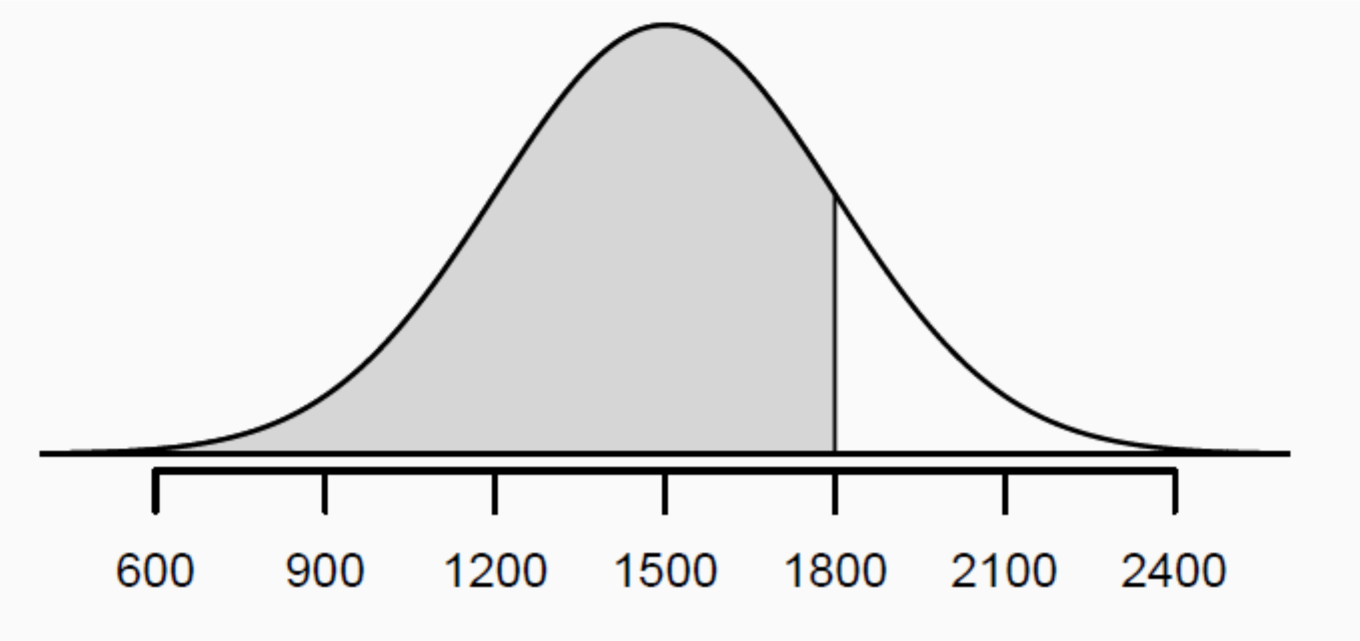


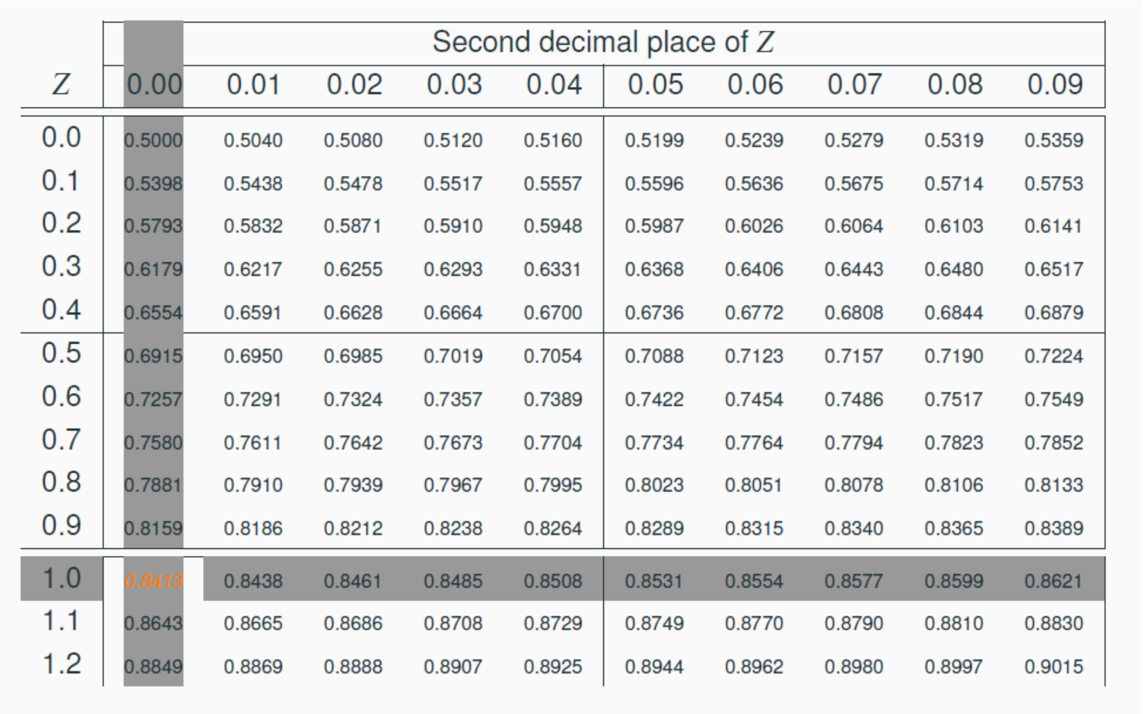
$$Z = \frac{observation - mean}{SD}$$

- Percentile is the percentage of observations that fall below a given data point.
- Graphically, percentile is the area below the probability distribution curve to the left of that observation.

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Standardizing with Z Scores



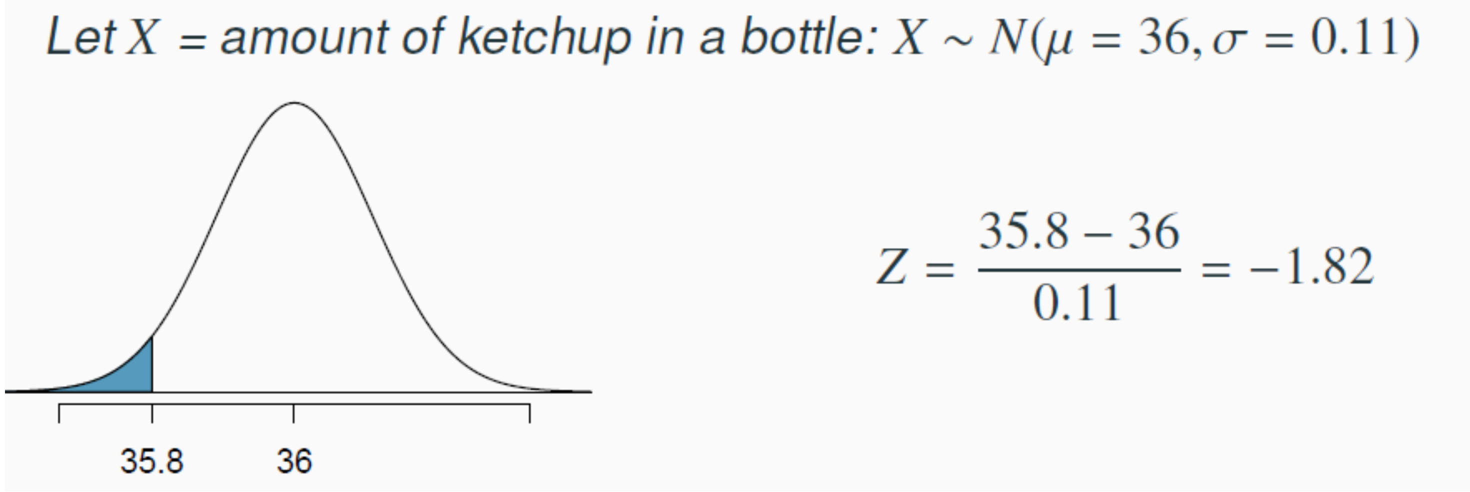


	Second decimal place of Z									
Z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Standardizing with Z Scores

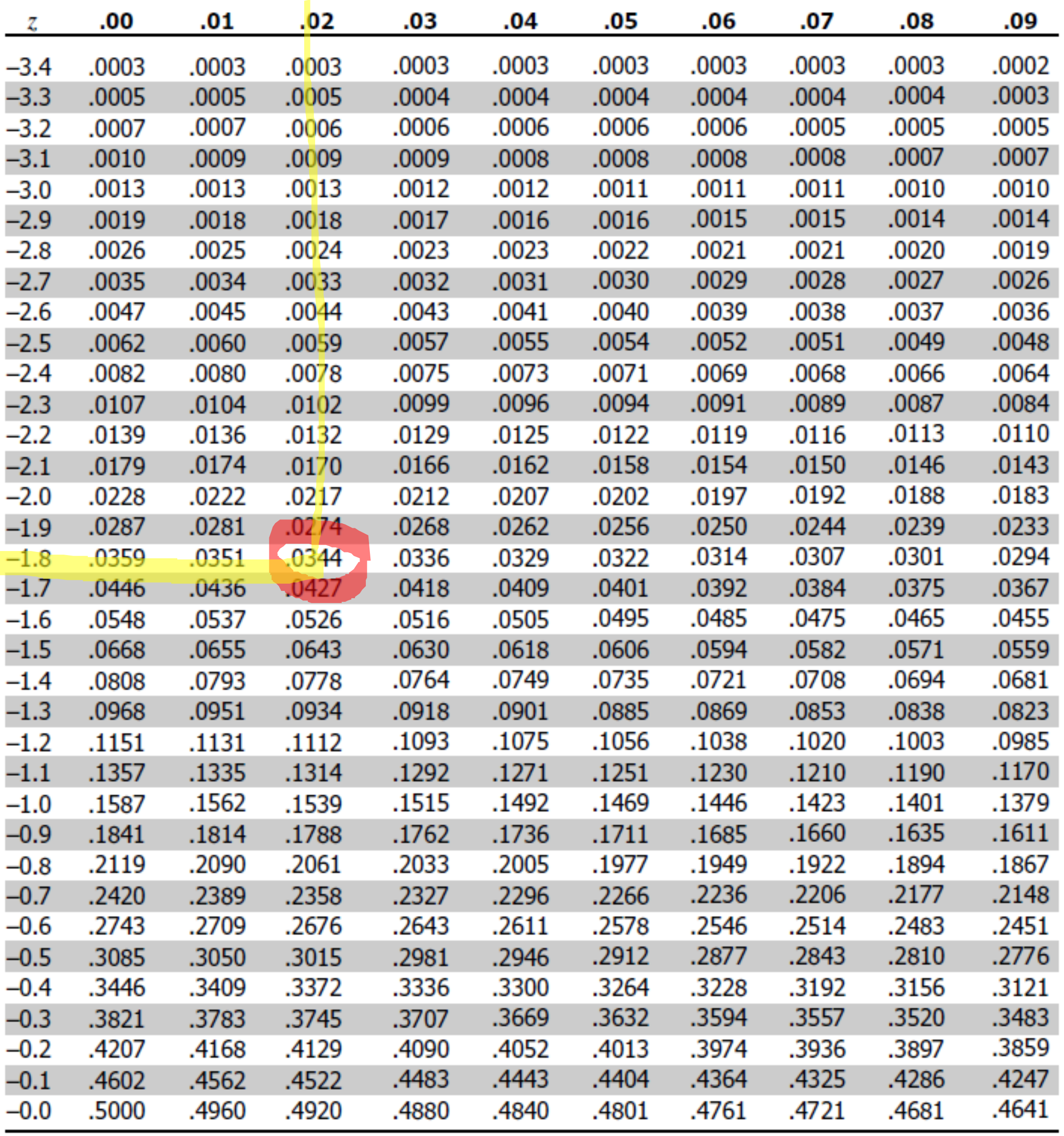
Example 2: At Heinz ketchup factory the amounts which go into bottles of ketchup are supposed to be normally distributed with mean 36 oz. and standard deviation 0.11 oz. Once every 30 minutes a bottle is selected from the production line, and its contents are noted precisely. If the amount of ketchup in the bottle is below 35.8 oz. or above 36.2 oz., then the bottle fails the quality control inspection. What percent of bottles have less than 35.8 ounces of ketchup?



Let X *= amount of ketchup in a bottle:* $X \sim N(\mu = 36, \sigma = 0.11)$

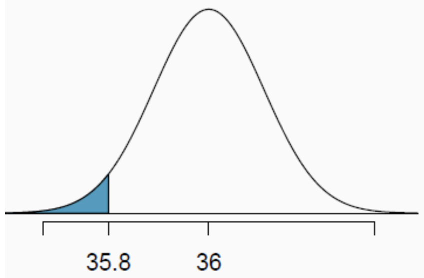
$$Z = \frac{35.8 - 36}{0.11} = -1.82$$

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel



z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
−3.4	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0002
−3.3	.0005	.0005	.0005	.0004	.0004	.0004	.0004	.0004	.0004	.0003
−3.2	.0007	.0007	.0006	.0006	.0006	.0006	.0006	.0005	.0005	.0005
−3.1	.0010	.0009	.0009	.0009	.0008	.0008	.0008	.0008	.0007	.0007
−3.0	.0013	.0013	.0013	.0012	.0012	.0011	.0011	.0011	.0010	.0010
−2.9	.0019	.0018	.0018	.0017	.0016	.0016	.0015	.0015	.0014	.0014
−2.8	.0026	.0025	.0024	.0023	.0023	.0022	.0021	.0021	.0020	.0019
−2.7	.0035	.0034	.0033	.0032	.0031	.0030	.0029	.0028	.0027	.0026
−2.6	.0047	.0045	.0044	.0043	.0041	.0040	.0039	.0038	.0037	.0036
−2.5	.0062	.0060	.0059	.0057	.0055	.0054	.0052	.0051	.0049	.0048
−2.4	.0082	.0080	.0078	.0075	.0073	.0071	.0069	.0068	.0066	.0064
−2.3	.0107	.0104	.0102	.0099	.0096	.0094	.0091	.0089	.0087	.0084
−2.2	.0139	.0136	.0132	.0129	.0125	.0122	.0119	.0116	.0113	.0110
−2.1	.0179	.0174	.0170	.0166	.0162	.0158	.0154	.0150	.0146	.0143
−2.0	.0228	.0222	.0217	.0212	.0207	.0202	.0197	.0192	.0188	.0183
−1.9	.0287	.0281	.0274	.0268	.0262	.0256	.0250	.0244	.0239	.0233
−1.8	.0359	.0351	.0344	.0336	.0329	.0322	.0314	.0307	.0301	.0294
−1.7	.0446	.0436	.0427	.0418	.0409	.0401	.0392	.0384	.0375	.0367
−1.6	.0548	.0537	.0526	.0516	.0505	.0495	.0485	.0475	.0465	.0455
−1.5	.0668	.0655	.0643	.0630	.0618	.0606	.0594	.0582	.0571	.0559
−1.4	.0808	.0793	.0778	.0764	.0749	.0735	.0721	.0708	.0694	.0681
−1.3	.0968	.0951	.0934	.0918	.0901	.0885	.0869	.0853	.0838	.0823
−1.2	.1151	.1131	.1112	.1093	.1075	.1056	.1038	.1020	.1003	.0985
−1.1	.1357	.1335	.1314	.1292	.1271	.1251	.1230	.1210	.1190	.1170
−1.0	.1587	.1562	.1539	.1515	.1492	.1469	.1446	.1423	.1401	.1379
−0.9	.1841	.1814	.1788	.1762	.1736	.1711	.1685	.1660	.1635	.1611
−0.8	.2119	.2090	.2061	.2033	.2005	.1977	.1949	.1922	.1894	.1867
−0.7	.2420	.2389	.2358	.2327	.2296	.2266	.2236	.2206	.2177	.2148
−0.6	.2743	.2709	.2676	.2643	.2611	.2578	.2546	.2514	.2483	.2451
−0.5	.3085	.3050	.3015	.2981	.2946	.2912	.2877	.2843	.2810	.2776
−0.4	.3446	.3409	.3372	.3336	.3300	.3264	.3228	.3192	.3156	.3121
−0.3	.3821	.3783	.3745	.3707	.3669	.3632	.3594	.3557	.3520	.3483
−0.2	.4207	.4168	.4129	.4090	.4052	.4013	.3974	.3936	.3897	.3859
−0.1	.4602	.4562	.4522	.4483	.4443	.4404	.4364	.4325	.4286	.4247
−0.0	.5000	.4960	.4920	.4880	.4840	.4801	.4761	.4721	.4681	.4641

Let X *= amount of ketchup in a bottle:* $X \sim N(\mu = 36, \sigma = 0.11)$

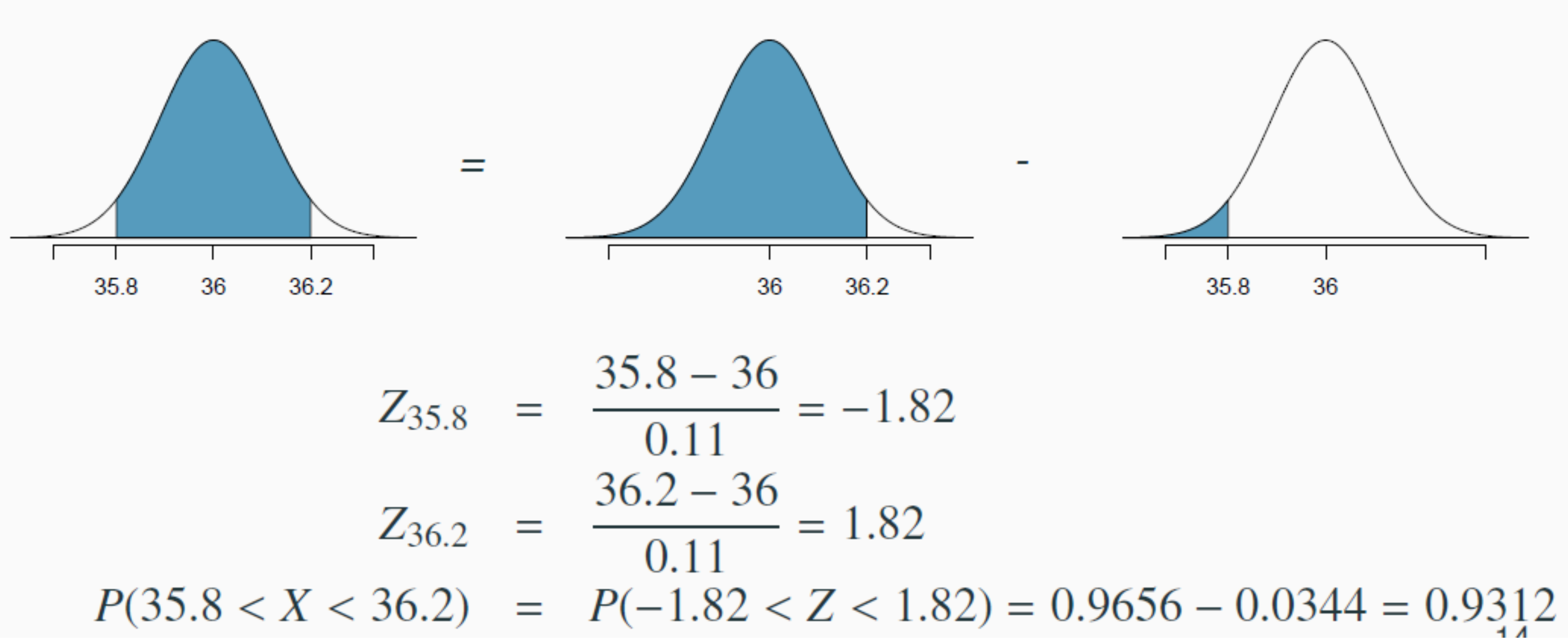


$$Z = \frac{35.8 - 36}{0.11} = -1.82$$

0.0344

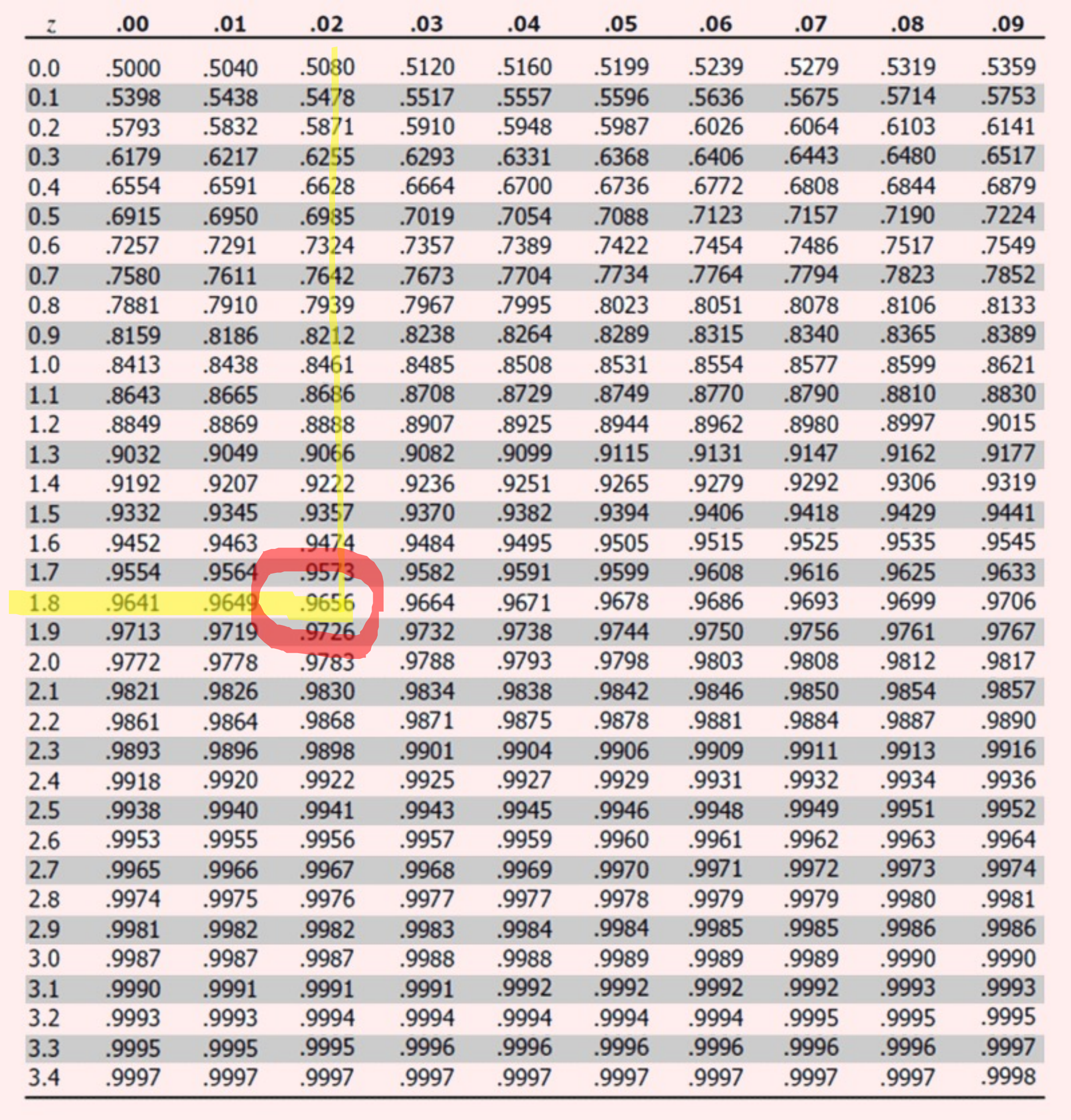
Standardizing with Z Scores

What percent of bottles pass the quality control inspection?



$$
\begin{aligned}
Z_{35.8} &= \frac{35.8 - 36}{0.11} = -1.82 \\
Z_{36.2} &= \frac{36.2 - 36}{0.11} = 1.82 \\
P(35.8 < X < 36.2) &= P(-1.82 < Z < 1.82) = 0.9656 - 0.0344 = 0.9312
\end{aligned}
$$

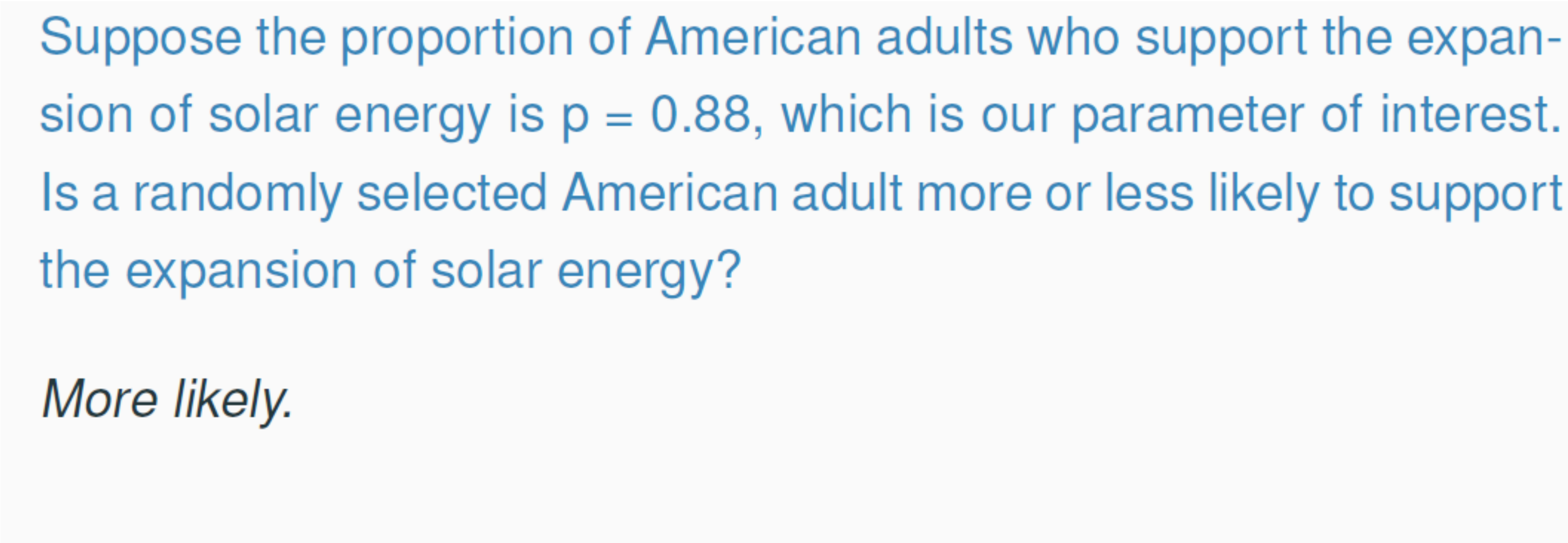
This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel



z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	.5000	.5040	.5080	.5120	.5160	.5199	.5239	.5279	.5319	.5359
0.1	.5398	.5438	.5478	.5517	.5557	.5596	.5636	.5675	.5714	.5753
0.2	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
0.3	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
0.4	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
0.5	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
0.6	.7257	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7517	.7549
0.7	.7580	.7611	.7642	.7673	.7704	.7734	.7764	.7794	.7823	.7852
0.8	.7881	.7910	.7939	.7967	.7995	.8023	.8051	.8078	.8106	.8133
0.9	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621
1.1	.8643	.8665	.8686	.8708	.8729	.8749	.8770	.8790	.8810	.8830
1.2	.8849	.8869	.8888	.8907	.8925	.8944	.8962	.8980	.8997	.9015
1.3	.9032	.9049	.9066	.9082	.9099	.9115	.9131	.9147	.9162	.9177
1.4	.9192	.9207	.9222	.9236	.9251	.9265	.9279	.9292	.9306	.9319
1.5	.9332	.9345	.9357	.9370	.9382	.9394	.9406	.9418	.9429	.9441
1.6	.9452	.9463	.9474	.9484	.9495	.9505	.9515	.9525	.9535	.9545
1.7	.9554	.9564	.9573	.9582	.9591	.9599	.9608	.9616	.9625	.9633
1.8	.9641	.9649	.9656	.9664	.9671	.9678	.9686	.9693	.9699	.9706
1.9	.9713	.9719	.9726	.9732	.9738	.9744	.9750	.9756	.9761	.9767
2.0	.9772	.9778	.9783	.9788	.9793	.9798	.9803	.9808	.9812	.9817
2.1	.9821	.9826	.9830	.9834	.9838	.9842	.9846	.9850	.9854	.9857
2.2	.9861	.9864	.9868	.9871	.9875	.9878	.9881	.9884	.9887	.9890
2.3	.9893	.9896	.9898	.9901	.9904	.9906	.9909	.9911	.9913	.9916
2.4	.9918	.9920	.9922	.9925	.9927	.9929	.9931	.9932	.9934	.9936
2.5	.9938	.9940	.9941	.9943	.9945	.9946	.9948	.9949	.9951	.9952
2.6	.9953	.9955	.9956	.9957	.9959	.9960	.9961	.9962	.9963	.9964
2.7	.9965	.9966	.9967	.9968	.9969	.9970	.9971	.9972	.9973	.9974
2.8	.9974	.9975	.9976	.9977	.9977	.9978	.9979	.9979	.9980	.9981
2.9	.9981	.9982	.9982	.9983	.9984	.9984	.9985	.9985	.9986	.9986
3.0	.9987	.9987	.9987	.9988	.9988	.9989	.9989	.9989	.9990	.9990
3.1	.9990	.9991	.9991	.9991	.9992	.9992	.9992	.9992	.9993	.9993
3.2	.9993	.9993	.9994	.9994	.9994	.9994	.9994	.9995	.9995	.9995
3.3	.9995	.9995	.9995	.9996	.9996	.9996	.9996	.9996	.9996	.9997
3.4	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9998

Z table is obtained from https://360digitmg.com/z-table

Interval Estimation



Suppose the proportion of American adults who support the expansion of solar energy is p = 0.88, which is our parameter of interest. Is a randomly selected American adult more or less likely to support the expansion of solar energy?

More likely.

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

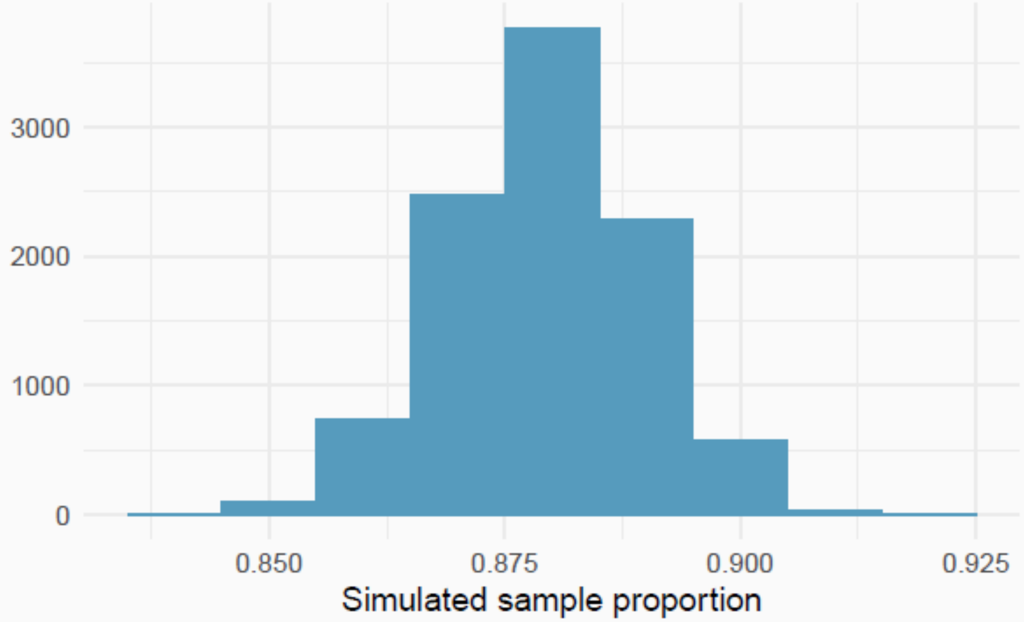
Suppose that you don’t have access to the populaion of all American adults, which is a quite likely scenario. In order to estimate the proportion of American adults who support solar power expansion, you might sample from the population and use your sample proportion as the best guess for the unknown population proportion.

- Sample, with replacement, 1000 American adults from the population, and record whether they support or not solar power expansion.
- Find the sample proportion.
- Plot the distribution of the sample proportions obtained by members of the class.

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

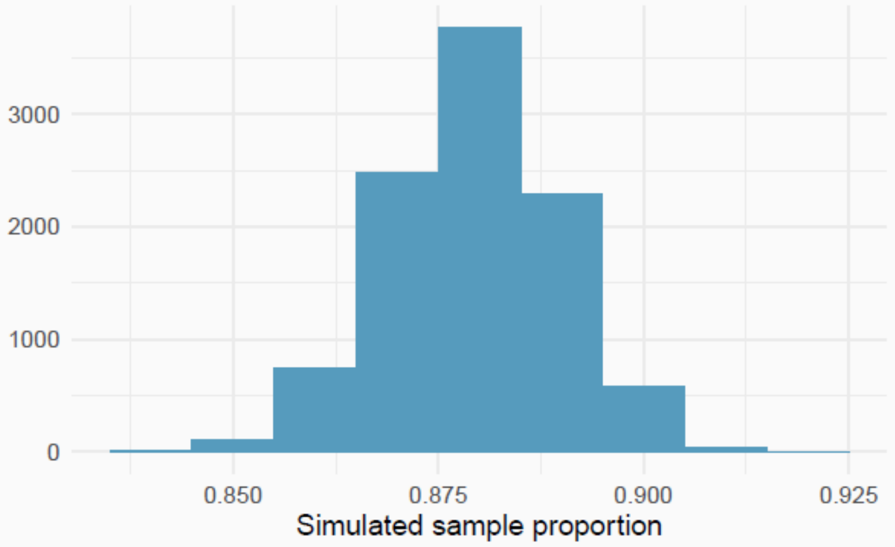
Sampling distribution

Suppose you were to repeat this process many times and obtain many $\hat{p}$s. This distribution is called a *sampling distribution*.



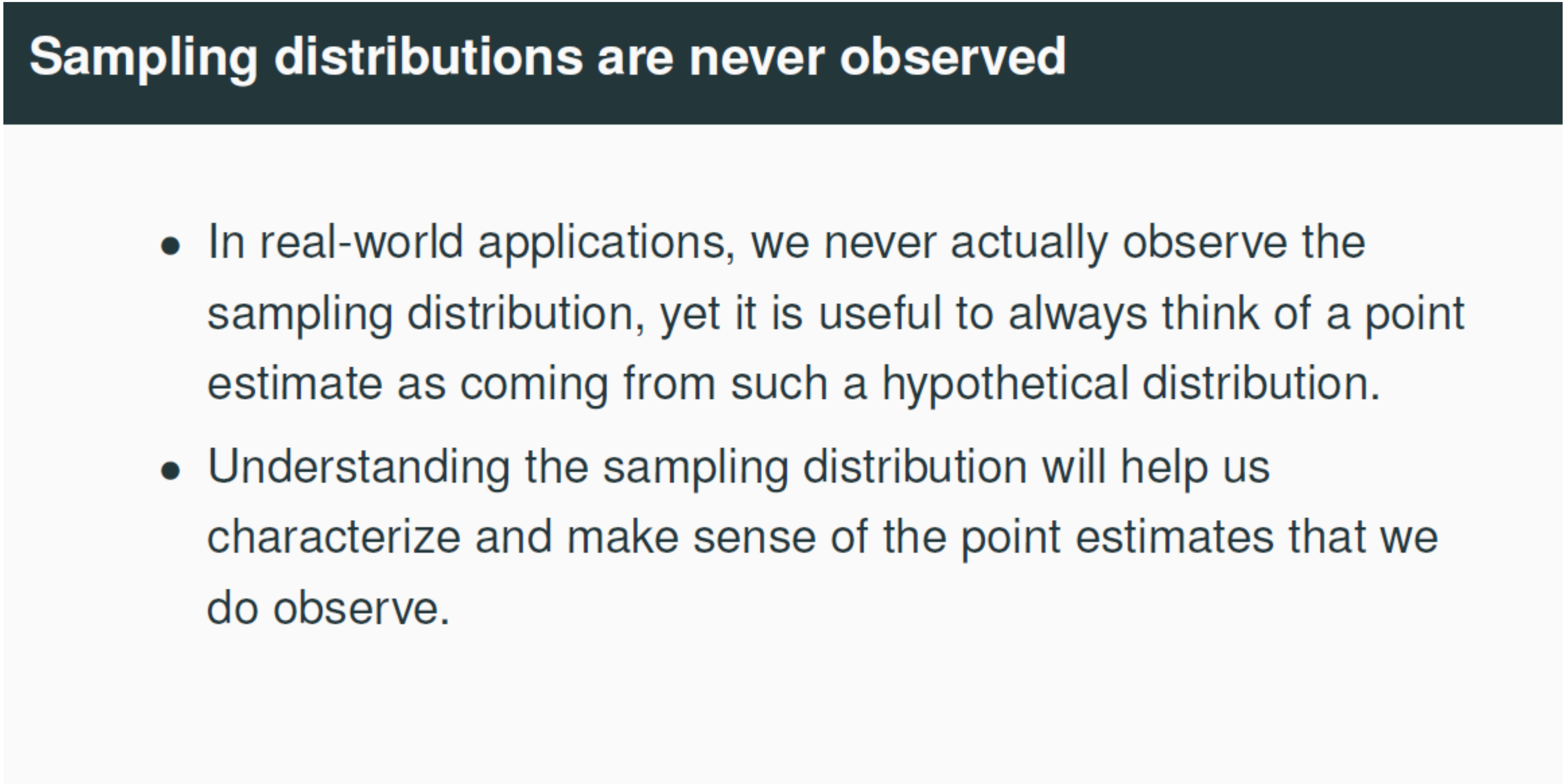
This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

What is the shape and center of this distribution? Based on this distribution, what do you think is the true population proportion?



The distribution is unimodal and roughly symmetric. A reasonable guess for the true population proportion is the center of this distribution, approximately 0.88.

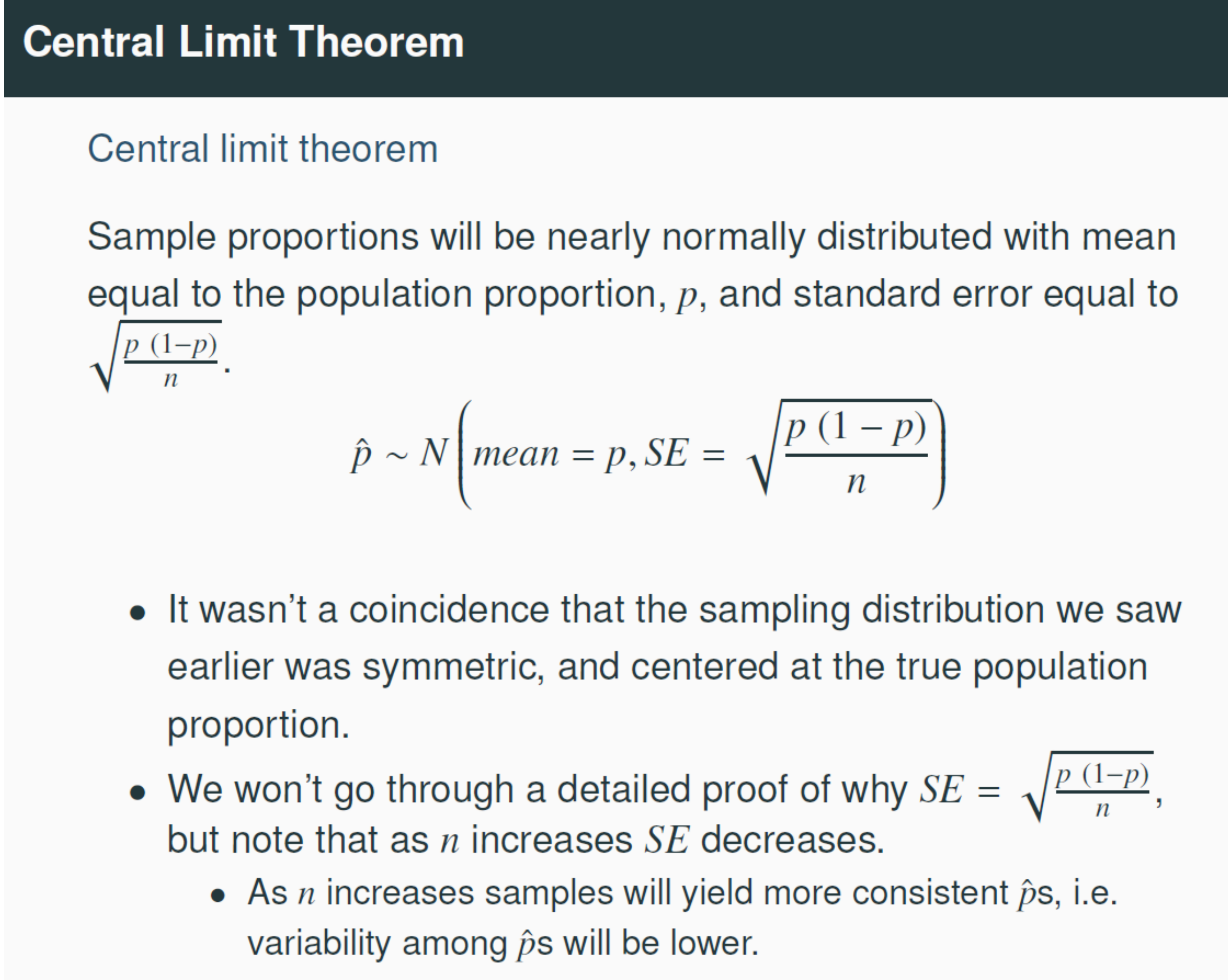
This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel



Sampling distributions are never observed

- In real-world applications, we never actually observe the sampling distribution, yet it is useful to always think of a point estimate as coming from such a hypothetical distribution.
- Understanding the sampling distribution will help us characterize and make sense of the point estimates that we do observe.

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel



Central Limit Theorem

Central limit theorem

Sample proportions will be nearly normally distributed with mean equal to the population proportion, p, and standard error equal to $\sqrt{\frac{p\,(1-p)}{n}}$.

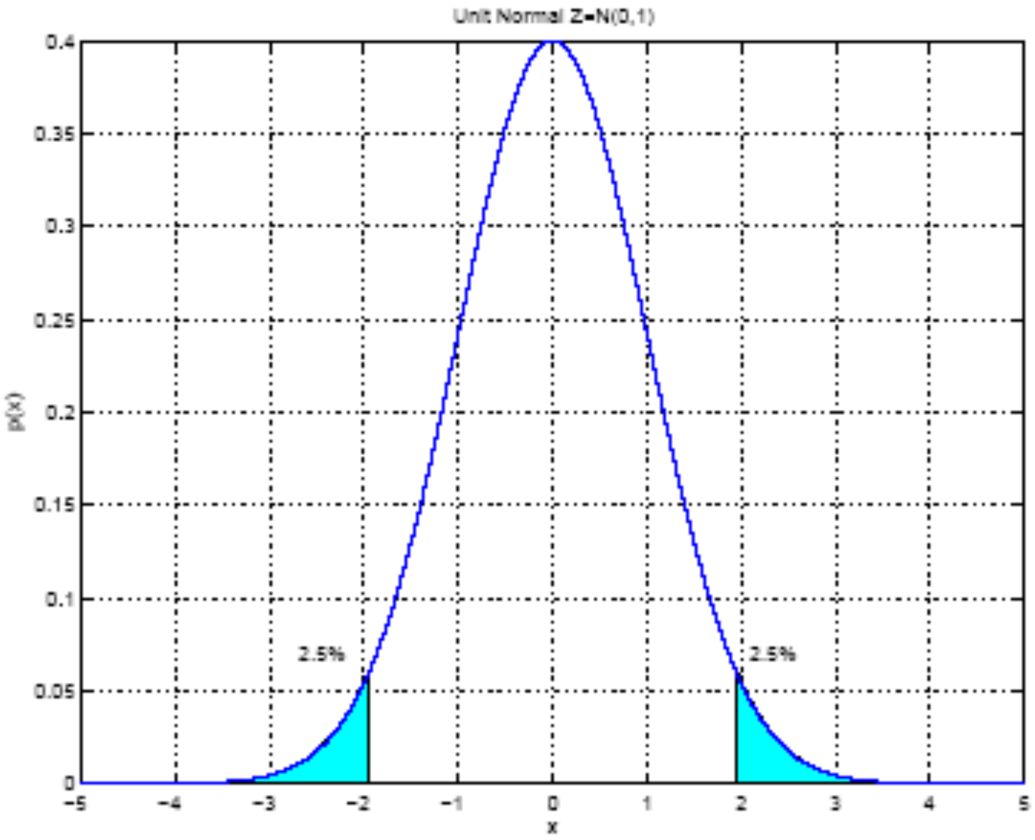
$$\hat{p} \sim N\left(mean = p, SE = \sqrt{\frac{p\,(1-p)}{n}}\right)$$

- It wasn’t a coincidence that the sampling distribution we saw earlier was symmetric, and centered at the true population proportion.
- We won’t go through a detailed proof of why $SE = \sqrt{\frac{p\,(1-p)}{n}}$, but note that as n increases SE decreases.
  - As n increases samples will yield more consistent $\hat{p}$s, i.e. variability among $\hat{p}$s will be lower.

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Interval Estimation

- $X = \{ x^t \}_t$ where $x^t \sim N(\mu, \sigma^2)$
- $m \sim N(\mu, \sigma^2/N)$



100(1- α) percent confidence interval

$$\sqrt{N}\frac{(m-\mu)}{\sigma} \sim Z$$

$$P\left\{-1.96 < \sqrt{N}\frac{(m-\mu)}{\sigma} < 1.96\right\} = 0.95$$

$$P\left\{m - 1.96\frac{\sigma}{\sqrt{N}} < \mu < m + 1.96\frac{\sigma}{\sqrt{N}}\right\} = 0.95$$

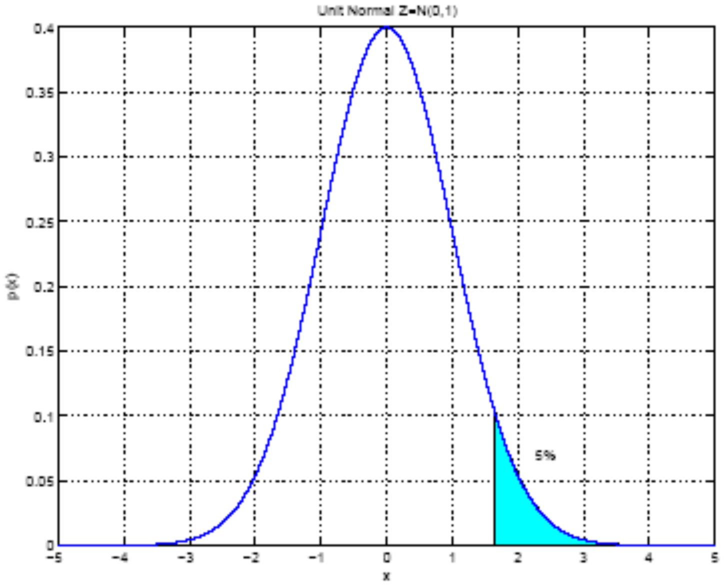
$$P\left\{m - z_{\alpha/2}\frac{\sigma}{\sqrt{N}} < \mu < m + z_{\alpha/2}\frac{\sigma}{\sqrt{N}}\right\} = 1-\alpha$$

- That is, “with 95 percent confidence,” μ will lie within
- $1.96\sigma/\sqrt{N}$ units of the sample average.

$$P\left\{\sqrt{N}\frac{(m-\mu)}{\sigma} < 1.64\right\} = 0.95$$

$$P\left\{m - 1.64\frac{\sigma}{\sqrt{N}} < \mu\right\} = 0.95$$

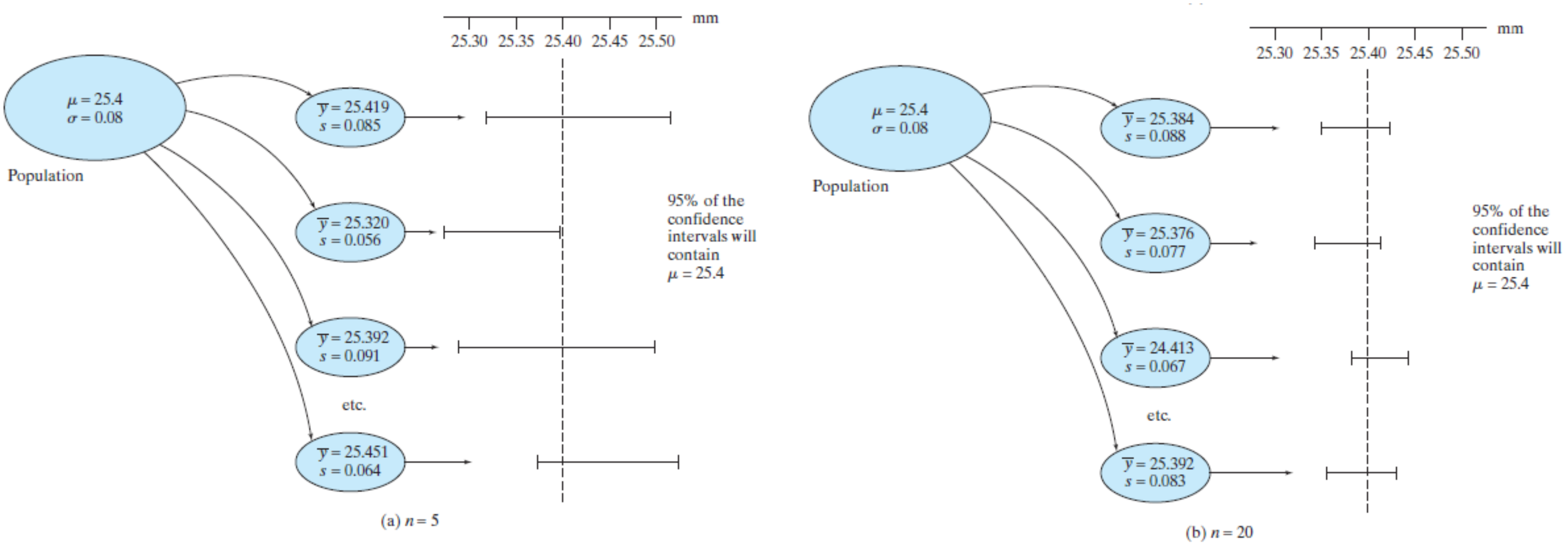
$$P\left\{m - z_{\alpha}\frac{\sigma}{\sqrt{N}} < \mu\right\} = 1 - \alpha$$



When σ^2 is not known:

$$S^2 = \sum_t \left(x^t - m\right)^2 / (N-1) \qquad \frac{\sqrt{N}(m-\mu)}{S} \sim t_{N-1}$$

$$P\left\{m - t_{\alpha/2,N-1}\frac{S}{\sqrt{N}} < \mu < m + t_{\alpha/2,N-1}\frac{S}{\sqrt{N}}\right\} = 1 - \alpha$$



(a) $n = 5$

(b) $n = 20$

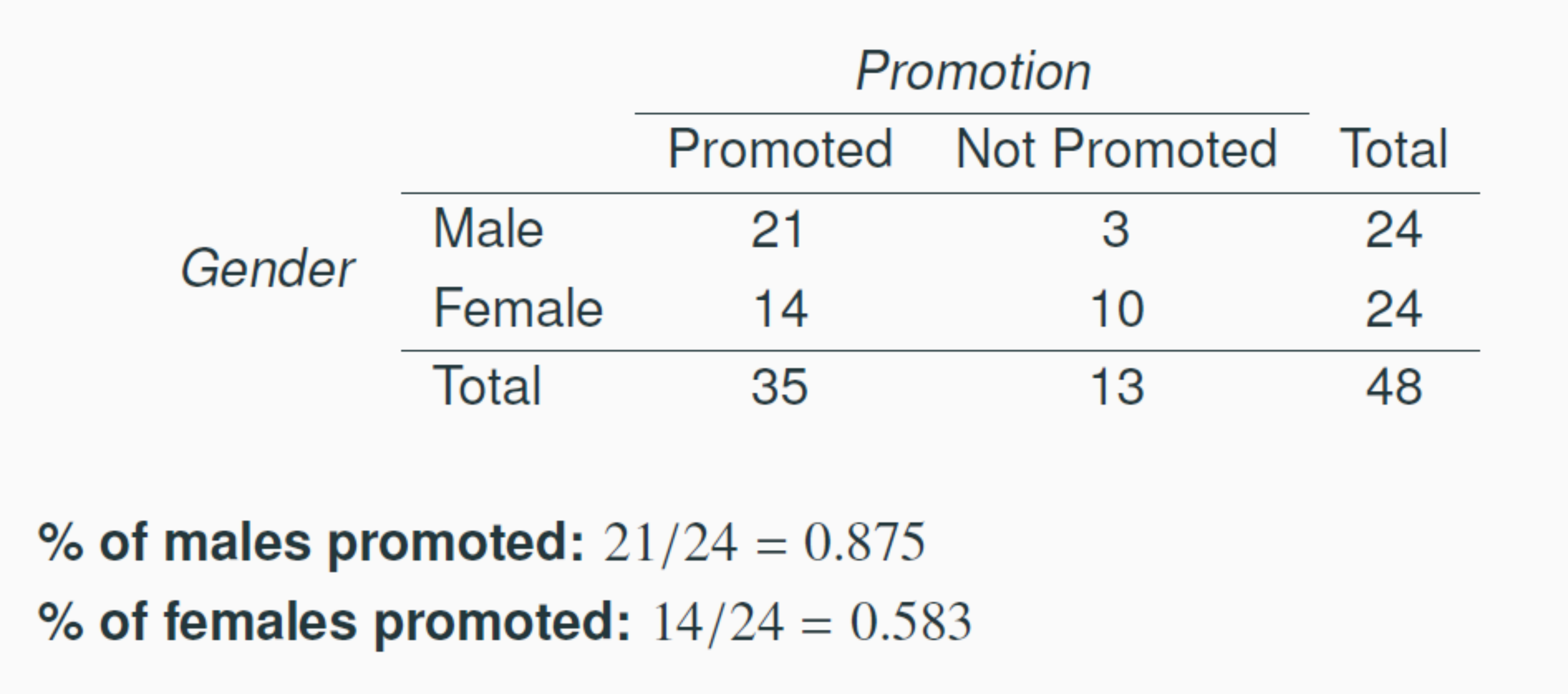
Adopted from the following book: Statistics for the Life Sciences, 5td Edition by Myra L. Samuels, Jeffrey A. Witmer, Andrew A. Schaffner

Hypothesis Testing

- In 1972, as a part of a study on gender discrimination, 48 male bank supervisors were each given the same personnel file and asked to judge whether the person should be promoted to a branch manager job that was described as "routine".
- The files were identical except that half of the supervisors had files showing the person was male while the other half had files showing the person was female
- It was randomly determined which supervisors got "male" applications and which got "female" applications.
- Of the 48 files reviewed, 35 were promoted.
- The study is testing whether females are unfairly discriminated against.

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Hypothesis Testing



		Promotion		
		Promoted	Not Promoted	Total
Gender	Male	21	3	24
	Female	14	10	24
	Total	35	13	48

% of males promoted: $21/24 = 0.875$

% of females promoted: $14/24 = 0.583$

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Hypothesis Testing

There is nothing going on."
Promotion and gender are independent, no gender discrimination, observed difference in proportions is simply due to chance:
Null hypothesis

"There is something going on."
Promotion and gender are dependent, there is gender discrimination, observed difference in proportions is not due to chance:
Alternative hypothesis

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Hypothesis Testing Framework

- We start with a null hypothesis (H_0) that represents the status quo.
- We also have an alternative hypothesis (H_A) that represents our research question, i.e. what we're testing for.

- We conduct a hypothesis test under the assumption that the null hypothesis is true, either via simulation or theoretical methods.

- If the test results suggest that the data do not provide convincing evidence for the alternative hypothesis, we stick with the null hypothesis.

If they do, then we reject the null hypothesis in favor of the alternative.

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

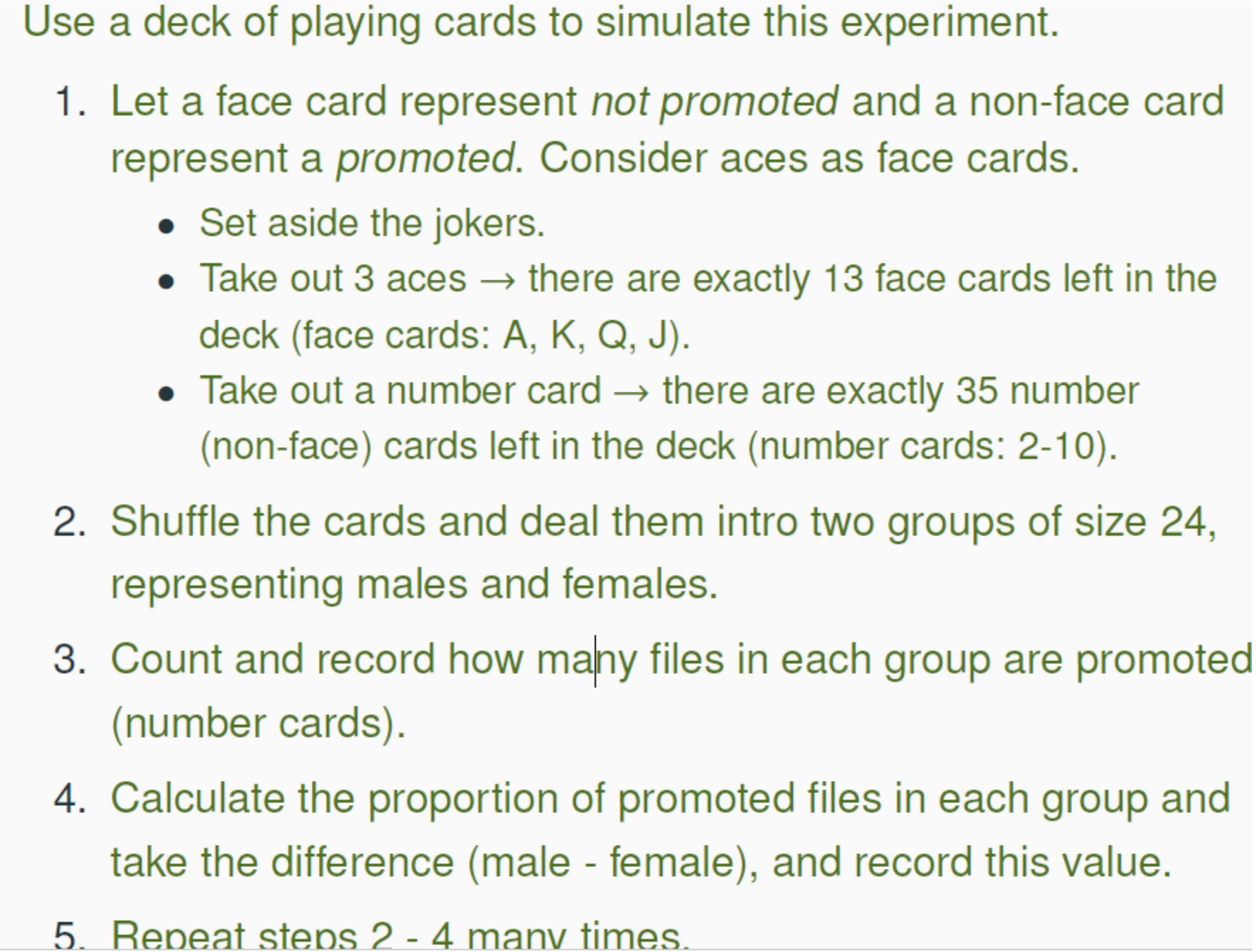
Hypothesis Testing Framework

If results from the simulations based on the chance model look like the data, then we can determine that the difference between the proportions of promoted files between males and females was simply due to chance (promotion and gender are independent).

If the results from the simulations based on the chance model do not look like the data, then we can determine that the difference between the proportions of promoted files between males and females was not due to chance, but due to an actual effect of gender (promotion and gender are dependent).

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Simulation

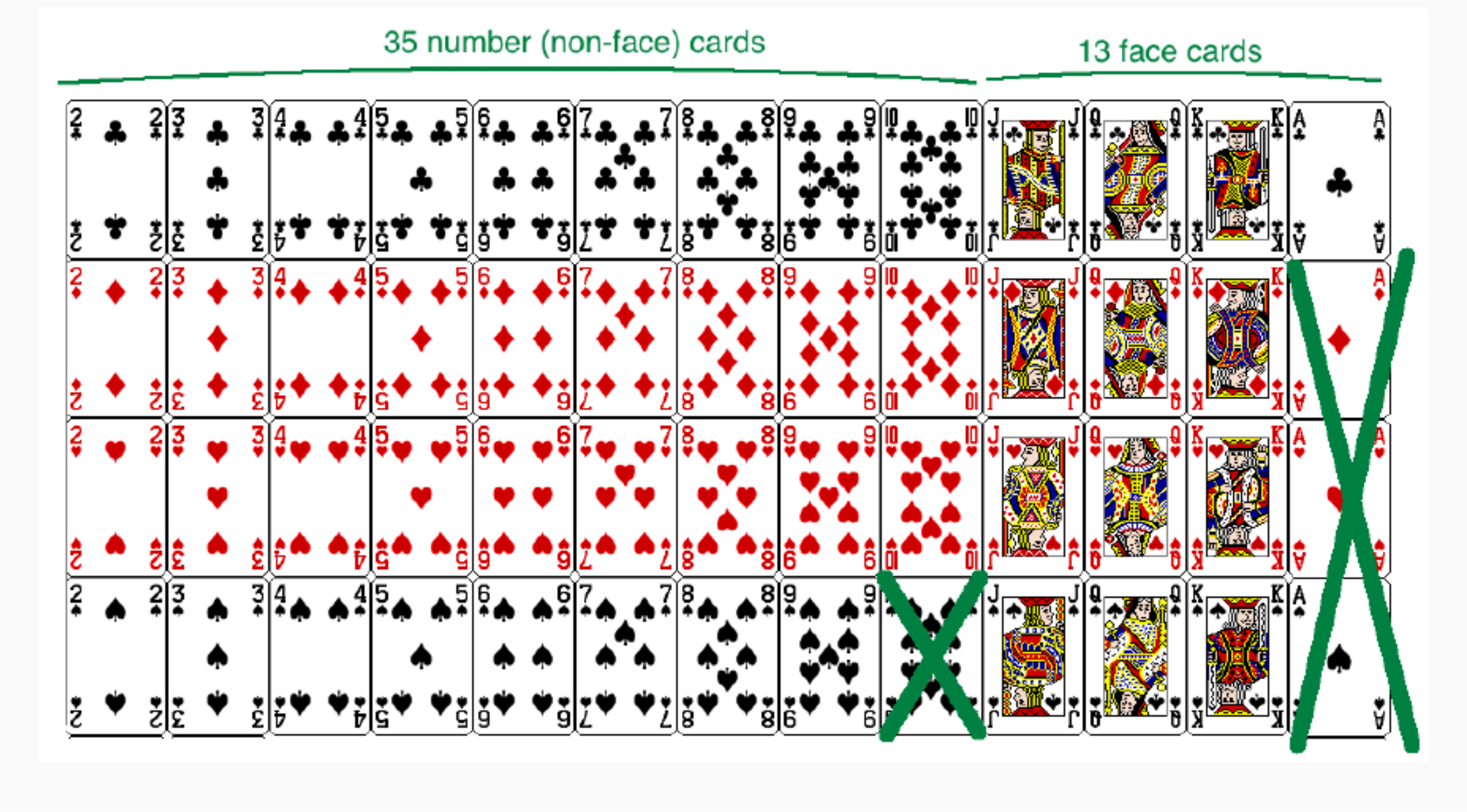


Use a deck of playing cards to simulate this experiment.

1. Let a face card represent *not promoted* and a non-face card represent a *promoted*. Consider aces as face cards.
   - Set aside the jokers.
   - Take out 3 aces → there are exactly 13 face cards left in the deck (face cards: A, K, Q, J).
   - Take out a number card → there are exactly 35 number (non-face) cards left in the deck (number cards: 2-10).
2. Shuffle the cards and deal them intro two groups of size 24, representing males and females.
3. Count and record how many files in each group are promoted (number cards).
4. Calculate the proportion of promoted files in each group and take the difference (male - female), and record this value.
5. Repeat steps 2 - 4 many times.

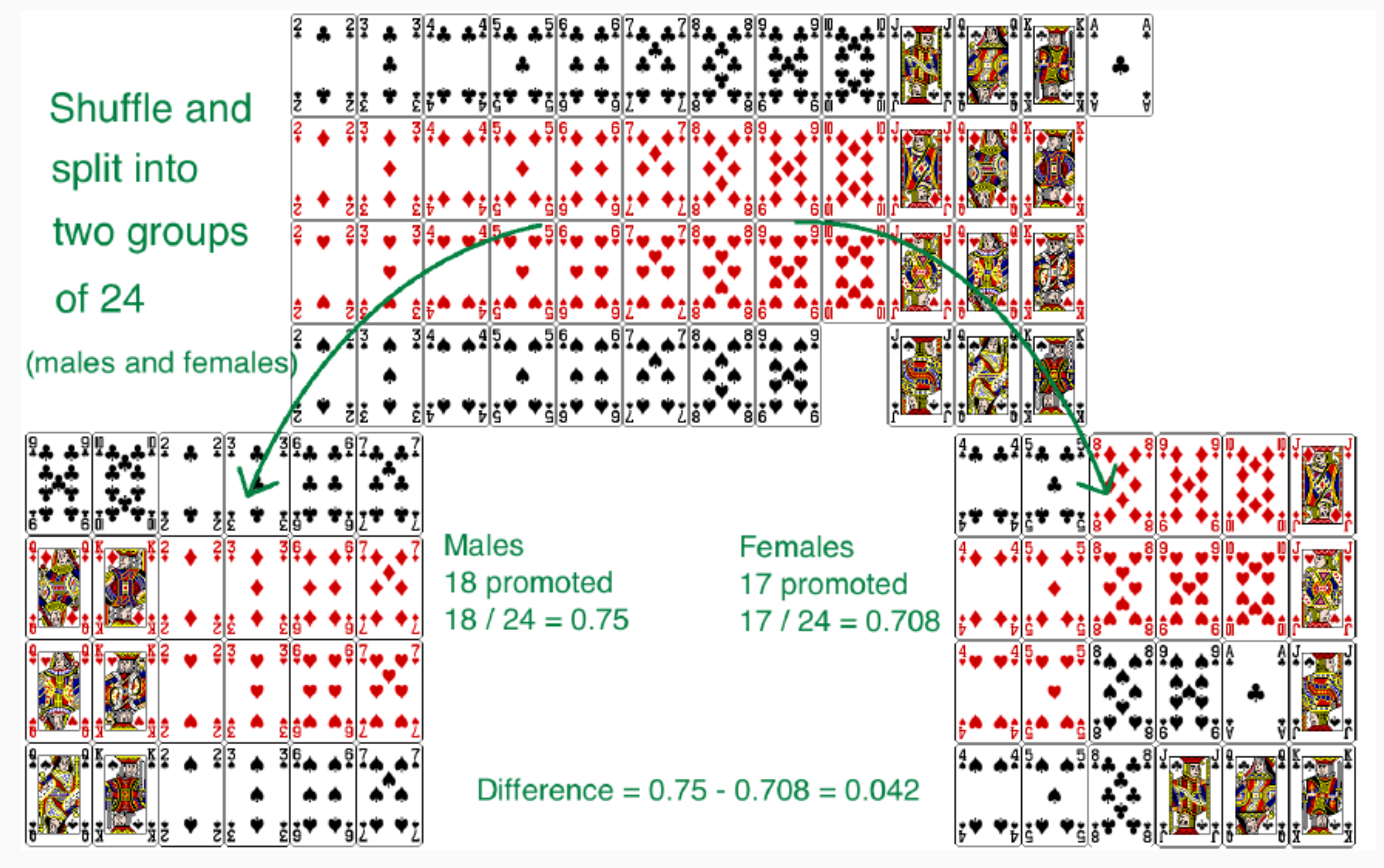
This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Simulation



This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Simulation



Shuffle and split into two groups of 24

(males and females)

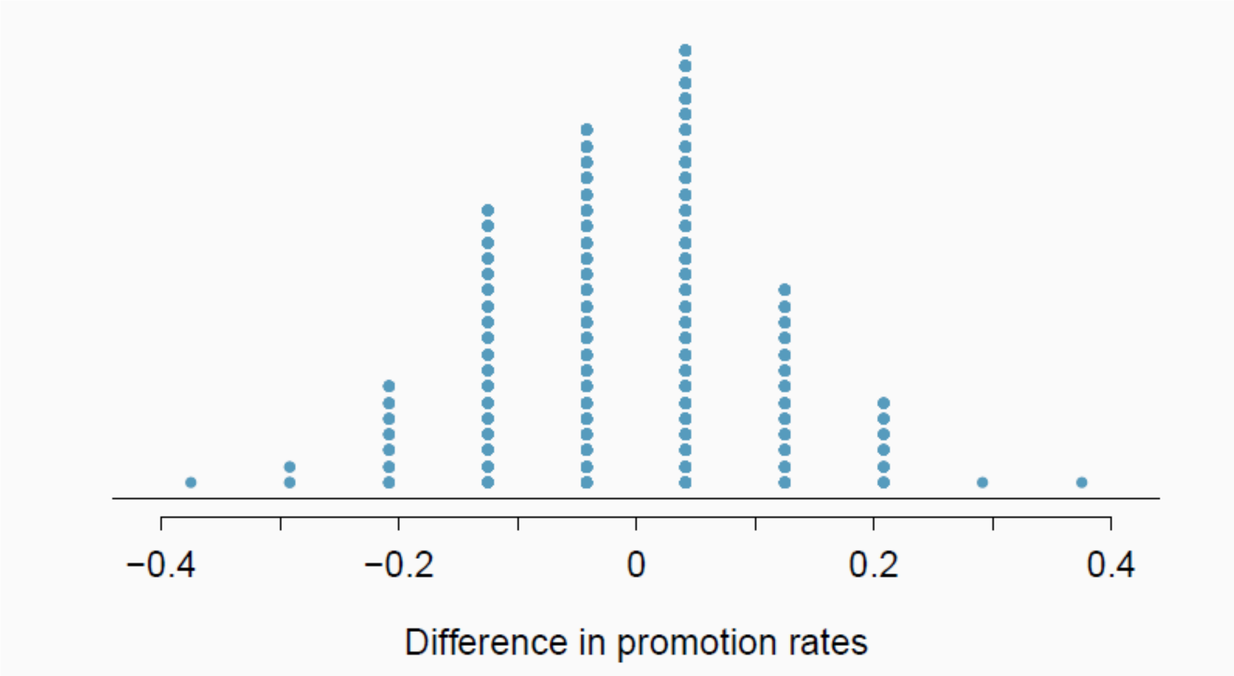
Males
18 promoted
18 / 24 = 0.75

Females
17 promoted
17 / 24 = 0.708

Difference = 0.75 - 0.708 = 0.042

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Simulation



The dot plot below shows the distribution of simulated differences in promotion rates based on 100 simulations.
The observed difference between the two proportions was due to a real effect of gender.

This slide is adopted from the following book: OpenIntro Statistics, 4th Edition by Mine Çetinkaya-Rundel

Hypothesis Testing

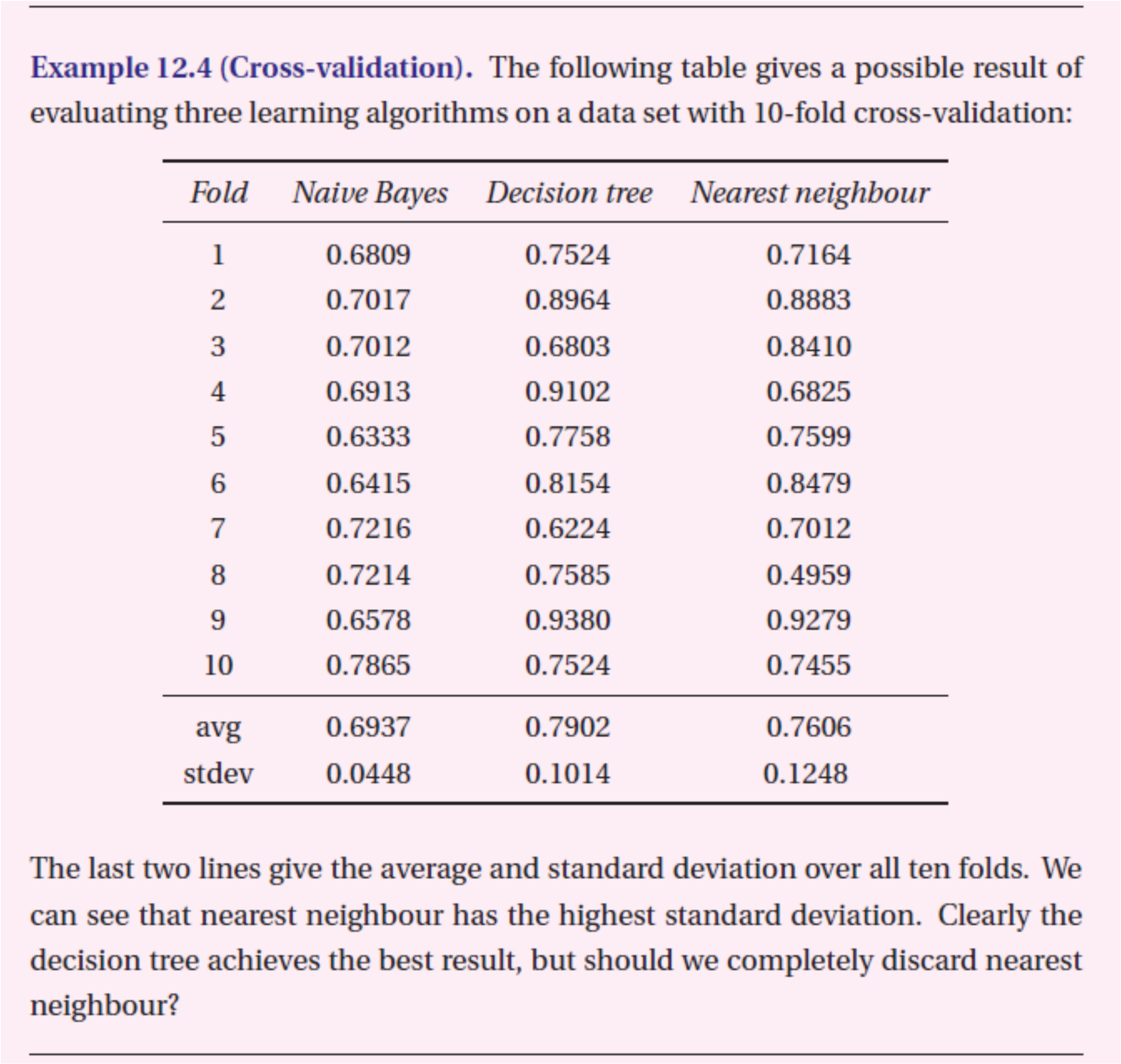
- Reject a null hypothesis if not supported by the sample with enough confidence
- X = { x^t }$_t$ where $x^t \sim$ N (μ, σ^2)

  H_0: $\mu = \mu_0$ vs. H_1: $\mu \neq \mu_0$

  Accept H_0 with level of significance α if μ_0 is in the 100(1- α) confidence interval

$$\frac{\sqrt{N}\left(m-\mu_0\right)}{\sigma} \in \left(-z_{\alpha/2}, z_{\alpha/2}\right)$$

Two-sided test

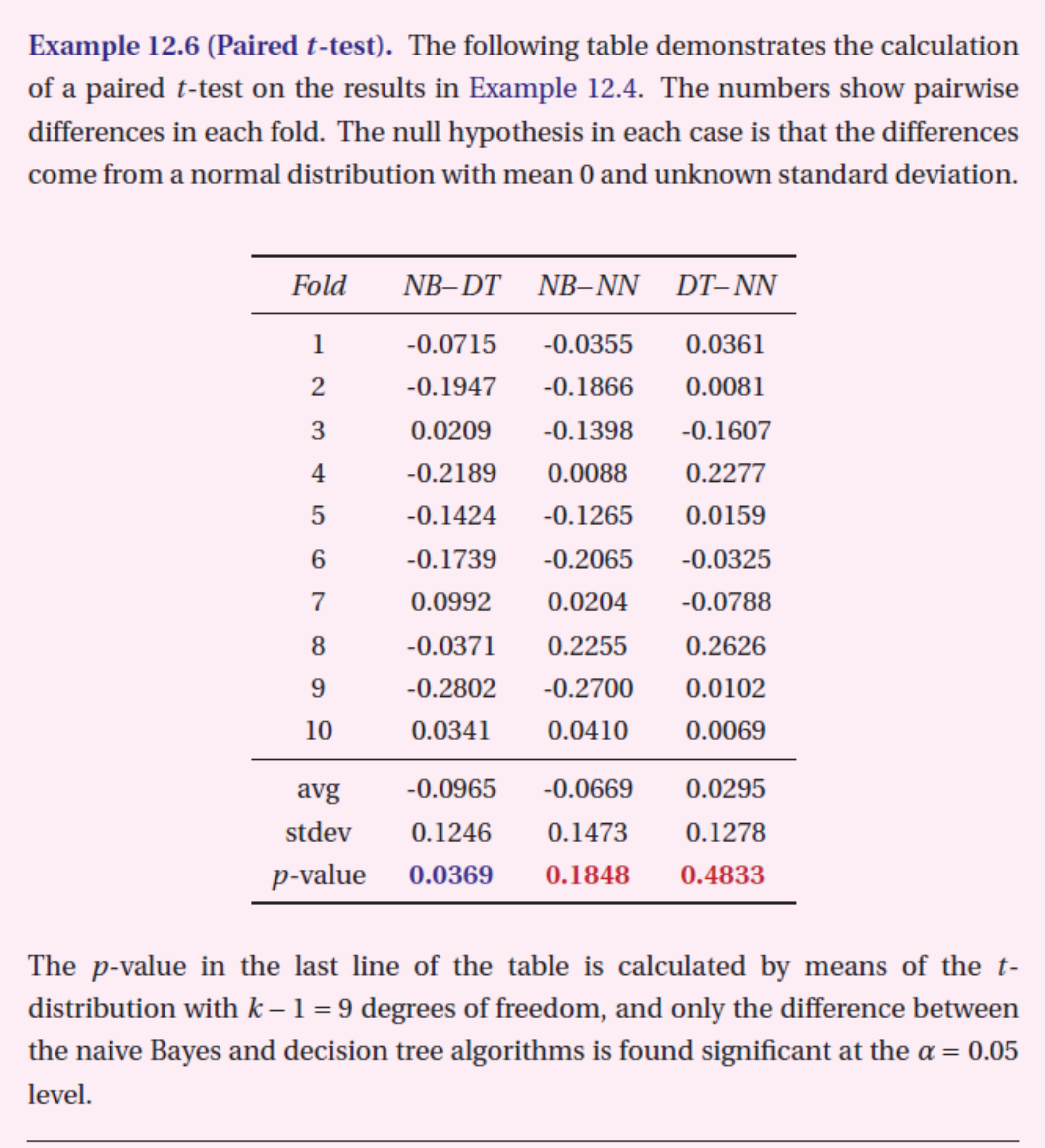


Example 12.4 (Cross-validation). The following table gives a possible result of evaluating three learning algorithms on a data set with 10-fold cross-validation:

Fold	*Naive Bayes*	*Decision tree*	*Nearest neighbour*
1	0.6809	0.7524	0.7164
2	0.7017	0.8964	0.8883
3	0.7012	0.6803	0.8410
4	0.6913	0.9102	0.6825
5	0.6333	0.7758	0.7599
6	0.6415	0.8154	0.8479
7	0.7216	0.6224	0.7012
8	0.7214	0.7585	0.4959
9	0.6578	0.9380	0.9279
10	0.7865	0.7524	0.7455
avg	0.6937	0.7902	0.7606
stdev	0.0448	0.1014	0.1248

The last two lines give the average and standard deviation over all ten folds. We can see that nearest neighbour has the highest standard deviation. Clearly the decision tree achieves the best result, but should we completely discard nearest neighbour?

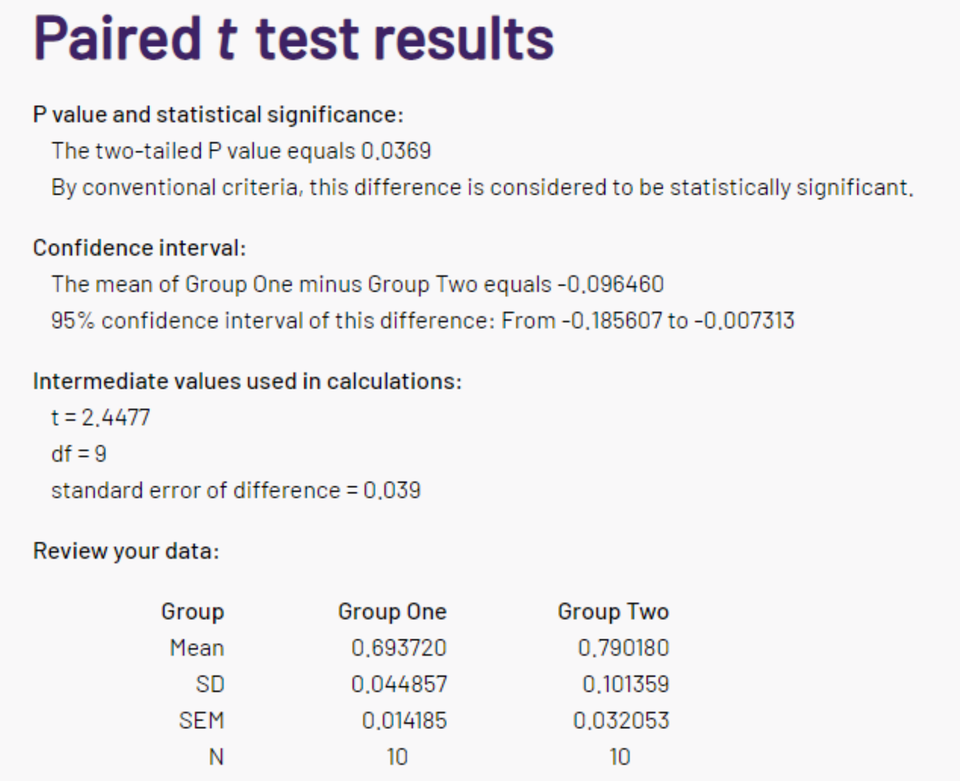
Adopted from «The Art and Science of Algorithms that Make Sense of Data», by Peter Flach



Example 12.6 (Paired t-test). The following table demonstrates the calculation of a paired t-test on the results in Example 12.4. The numbers show pairwise differences in each fold. The null hypothesis in each case is that the differences come from a normal distribution with mean 0 and unknown standard deviation.

Fold	*NB–DT*	*NB–NN*	*DT–NN*
1	-0.0715	-0.0355	0.0361
2	-0.1947	-0.1866	0.0081
3	0.0209	-0.1398	-0.1607
4	-0.2189	0.0088	0.2277
5	-0.1424	-0.1265	0.0159
6	-0.1739	-0.2065	-0.0325
7	0.0992	0.0204	-0.0788
8	-0.0371	0.2255	0.2626
9	-0.2802	-0.2700	0.0102
10	0.0341	0.0410	0.0069
avg	-0.0965	-0.0669	0.0295
stdev	0.1246	0.1473	0.1278
p-value	0.0369	0.1848	0.4833

The p-value in the last line of the table is calculated by means of the t-distribution with $k - 1 = 9$ degrees of freedom, and only the difference between the naive Bayes and decision tree algorithms is found significant at the $\alpha = 0.05$ level.



Paired *t* test results

P value and statistical significance:

The two-tailed P value equals 0.0369

By conventional criteria, this difference is considered to be statistically significant.

Confidence interval:

The mean of Group One minus Group Two equals -0.096460

95% confidence interval of this difference: From -0.185607 to -0.007313

Intermediate values used in calculations:

t = 2.4477

df = 9

standard error of difference = 0.039

Review your data:

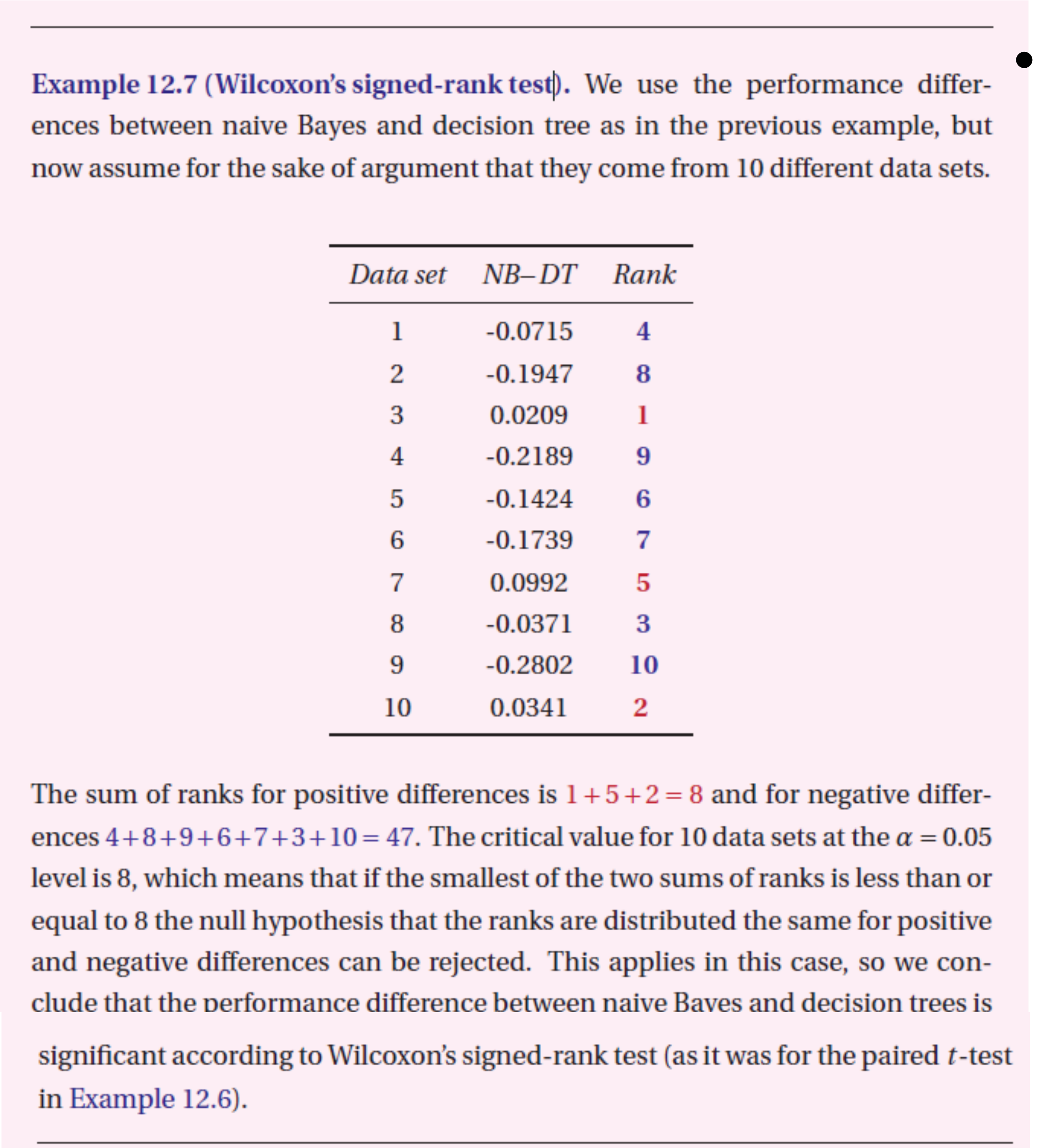
Group	Group One	Group Two
Mean	0.693720	0.790180
SD	0.044857	0.101359
SEM	0.014185	0.032053
N	10	10

Adopted from «The Art and Science of Algorithms that Make Sense of Data», by Peter Flach

- The *t* -test can be applied for comparing two learning algorithms over a single data set, typically using results obtained in cross-validation.
- It is not appropriate for multiple data sets because performance measures cannot be compared across data sets (they are not 'commensurate').
- In order to compare two learning algorithms over multiple data sets we need to use a test specifically designed for that purpose such as *Wilcoxon's signed-rank test*.

Wilcoxon’s signed-rank

- The idea is to rank the performance differences in absolute value, from smallest (rank 1) to largest (rank *n*).
- We then calculate the sum of ranks for positive and negative differences separately, and take the smaller of these sums as our test statistic.



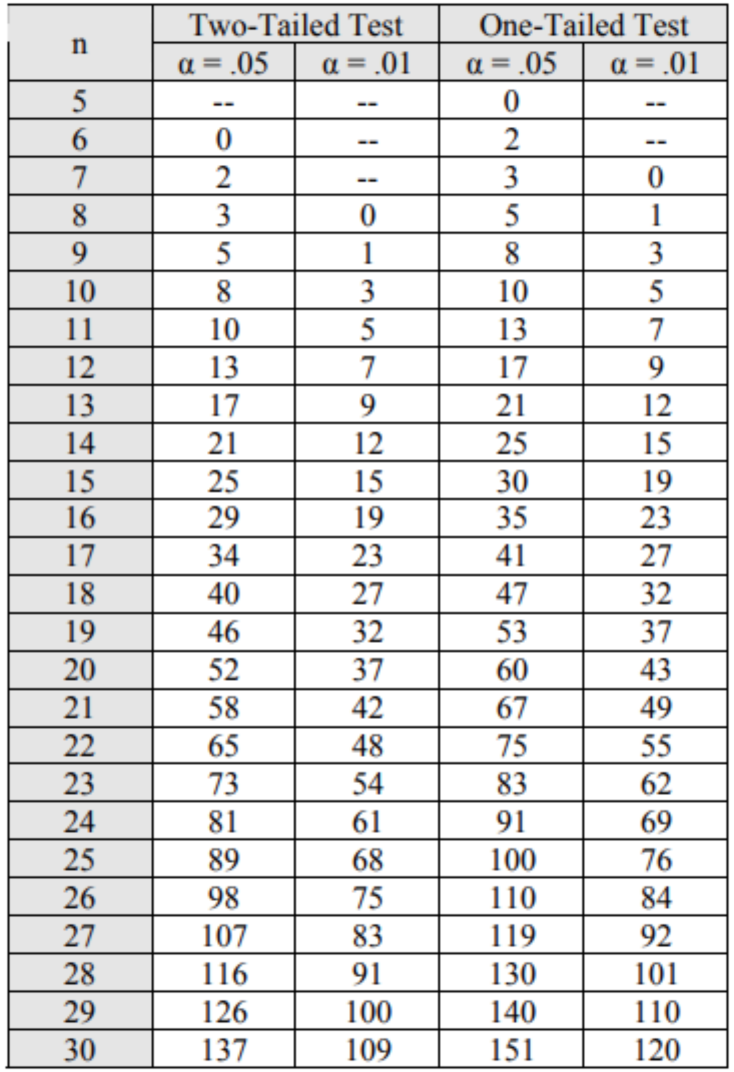
Example 12.7 (Wilcoxon's signed-rank test). We use the performance differences between naive Bayes and decision tree as in the previous example, but now assume for the sake of argument that they come from 10 different data sets.

Data set	*NB–DT*	*Rank*
1	-0.0715	4
2	-0.1947	8
3	0.0209	1
4	-0.2189	9
5	-0.1424	6
6	-0.1739	7
7	0.0992	5
8	-0.0371	3
9	-0.2802	10
10	0.0341	2

The sum of ranks for positive differences is $1+5+2=8$ and for negative differences $4+8+9+6+7+3+10=47$. The critical value for 10 data sets at the $\alpha=0.05$ level is 8, which means that if the smallest of the two sums of ranks is less than or equal to 8 the null hypothesis that the ranks are distributed the same for positive and negative differences can be rejected. This applies in this case, so we conclude that the performance difference between naive Bayes and decision trees is significant according to Wilcoxon's signed-rank test (as it was for the paired t-test in Example 12.6).

Critical Values of the Wilcoxon Signed Ranks Test

- The smallest of ranks are ≤ critical level we can reject Null Hypothesis

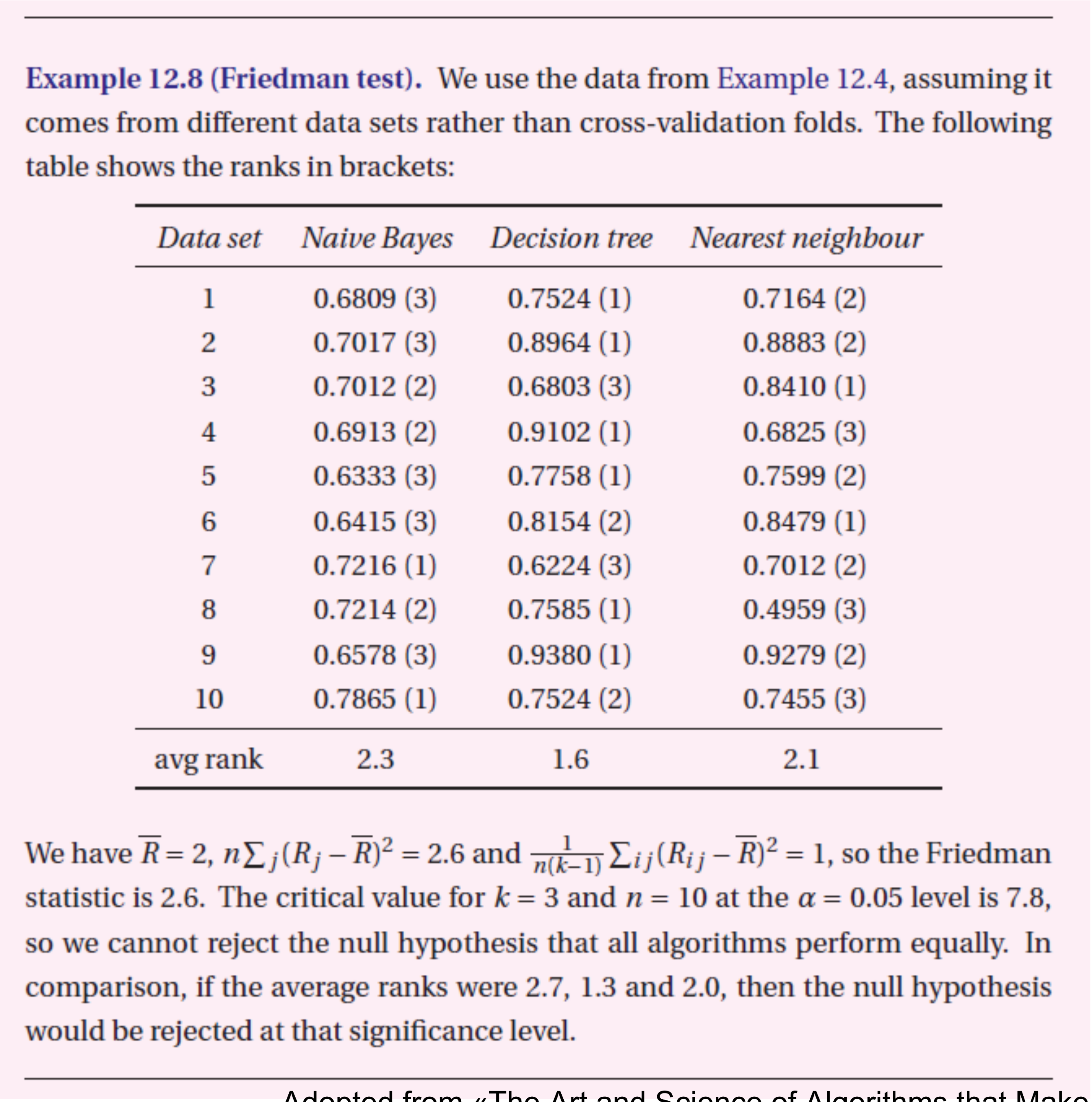


n	Two-Tailed Test		One-Tailed Test	
	α = .05	α = .01	α = .05	α = .01
5	--	--	0	--
6	0	--	2	--
7	2	--	3	0
8	3	0	5	1
9	5	1	8	3
10	8	3	10	5
11	10	5	13	7
12	13	7	17	9
13	17	9	21	12
14	21	12	25	15
15	25	15	30	19
16	29	19	35	23
17	34	23	41	27
18	40	27	47	32
19	46	32	53	37
20	52	37	60	43
21	58	42	67	49
22	65	48	75	55
23	73	54	83	62
24	81	61	91	69
25	89	68	100	76
26	98	75	110	84
27	107	83	119	92
28	116	91	130	101
29	126	100	140	110
30	137	109	151	120

Adopted from «The Art and Science of Algorithms that Make Sense of Data», by Peter Flach

Friedman Test

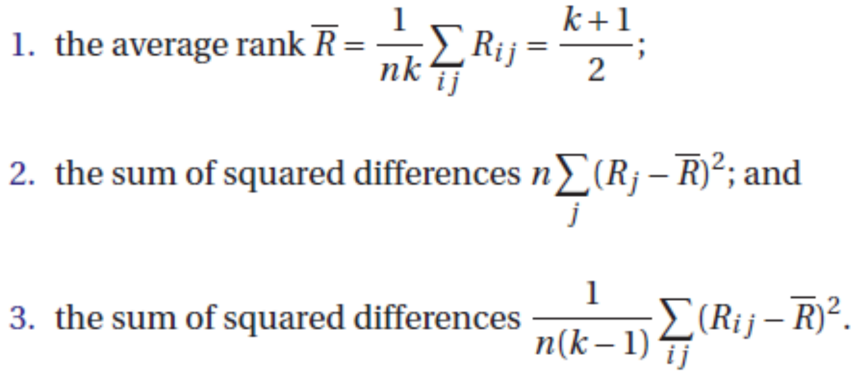
- If we want to compare *k* algorithms over *n* data sets we need to use specialized significance test.
- The idea is to rank the performance of all *k* algorithms per data set, from best performance (rank 1) to worst performance (rank *k*).
- Under the null hypothesis that all algorithms perform equally these average ranks *Rj* should be the same.
- In order to test this we calculate the following quantities:



Example 12.8 (Friedman test). We use the data from Example 12.4, assuming it comes from different data sets rather than cross-validation folds. The following table shows the ranks in brackets:

Data set	*Naive Bayes*	*Decision tree*	*Nearest neighbour*
1	0.6809 (3)	0.7524 (1)	0.7164 (2)
2	0.7017 (3)	0.8964 (1)	0.8883 (2)
3	0.7012 (2)	0.6803 (3)	0.8410 (1)
4	0.6913 (2)	0.9102 (1)	0.6825 (3)
5	0.6333 (3)	0.7758 (1)	0.7599 (2)
6	0.6415 (3)	0.8154 (2)	0.8479 (1)
7	0.7216 (1)	0.6224 (3)	0.7012 (2)
8	0.7214 (2)	0.7585 (1)	0.4959 (3)
9	0.6578 (3)	0.9380 (1)	0.9279 (2)
10	0.7865 (1)	0.7524 (2)	0.7455 (3)
avg rank	2.3	1.6	2.1

We have $\overline{R} = 2$, $n\sum_j (R_j - \overline{R})^2 = 2.6$ and $\frac{1}{n(k-1)}\sum_{ij}(R_{ij} - \overline{R})^2 = 1$, so the Friedman statistic is 2.6. The critical value for $k = 3$ and $n = 10$ at the $\alpha = 0.05$ level is 7.8, so we cannot reject the null hypothesis that all algorithms perform equally. In comparison, if the average ranks were 2.7, 1.3 and 2.0, then the null hypothesis would be rejected at that significance level.



1. the average rank $\overline{R} = \frac{1}{nk}\sum_{ij} R_{ij} = \frac{k+1}{2}$;
2. the sum of squared differences $n\sum_j (R_j - \overline{R})^2$; and
3. the sum of squared differences $\frac{1}{n(k-1)}\sum_{ij}(R_{ij} - \overline{R})^2$.

- If Friedman Statistics ≤ critical level we can reject Null Hypothesis

Adopted from «The Art and Science of Algorithms that Make Sense of Data», by Peter Flach

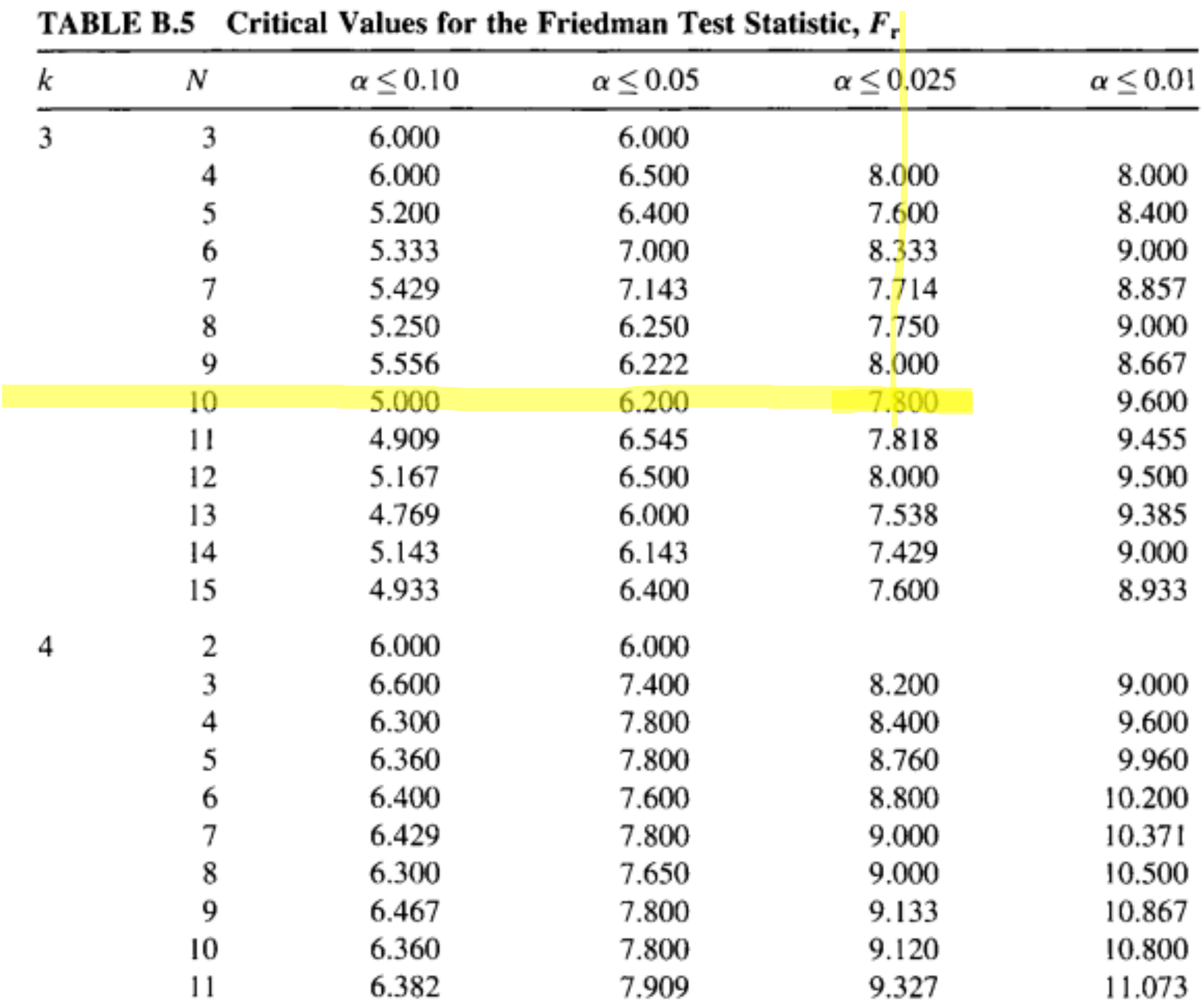


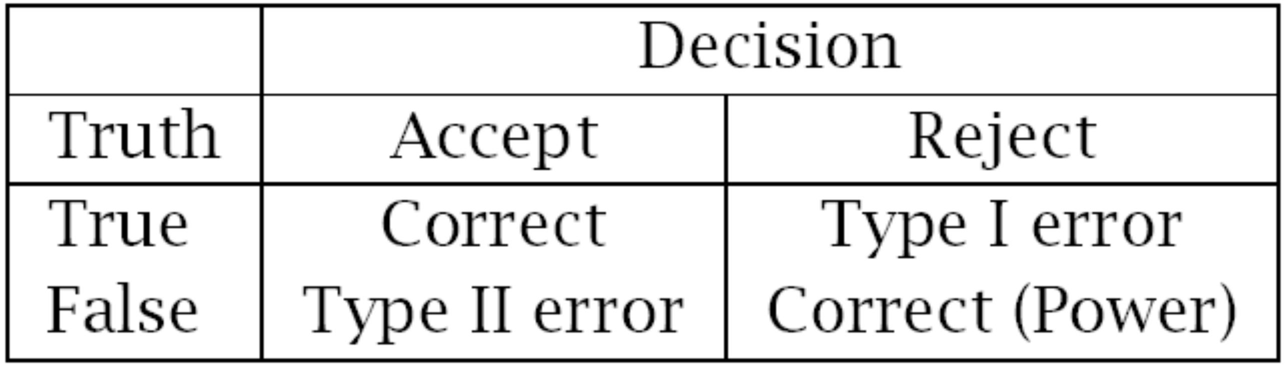
TABLE B.5 Critical Values for the Friedman Test Statistic, F_r

k	N	$\alpha \leq 0.10$	$\alpha \leq 0.05$	$\alpha \leq 0.025$	$\alpha \leq 0.01$
3	3	6.000	6.000		
	4	6.000	6.500	8.000	8.000
	5	5.200	6.400	7.600	8.400
	6	5.333	7.000	8.333	9.000
	7	5.429	7.143	7.714	8.857
	8	5.250	6.250	7.750	9.000
	9	5.556	6.222	8.000	8.667
	10	5.000	6.200	7.800	9.600
	11	4.909	6.545	7.818	9.455
	12	5.167	6.500	8.000	9.500
	13	4.769	6.000	7.538	9.385
	14	5.143	6.143	7.429	9.000
	15	4.933	6.400	7.600	8.933
4	2	6.000	6.000		
	3	6.600	7.400	8.200	9.000
	4	6.300	7.800	8.400	9.600
	5	6.360	7.800	8.760	9.960
	6	6.400	7.600	8.800	10.200
	7	6.429	7.800	9.000	10.371
	8	6.300	7.650	9.000	10.500
	9	6.467	7.800	9.133	10.867
	10	6.360	7.800	9.120	10.800
	11	6.382	7.909	9.327	11.073

ALPAYDIN’S SLIDES

IMPORTANT

Even if we fail to reject the null hypothesis, we typically do not accept the null hypothesis as true.



	Decision	
Truth	Accept	Reject
True	Correct	Type I error
False	Type II error	Correct (Power)

Which one is more crucial, Type I or II error?

- One-sided test: H_0: $\mu \leq \mu_0$ vs. H_1: $\mu > \mu_0$

  Accept if $\dfrac{\sqrt{N}(m-\mu_0)}{\sigma} \in (-\infty, z_\alpha)$

- Variance unknown: Use t, instead of z

  Accept H_0: $\mu = \mu_0$ if $\dfrac{\sqrt{N}(m-\mu_0)}{S} \in \left(-t_{\alpha/2,N-1}, t_{\alpha/2,N-1}\right)$

t Test

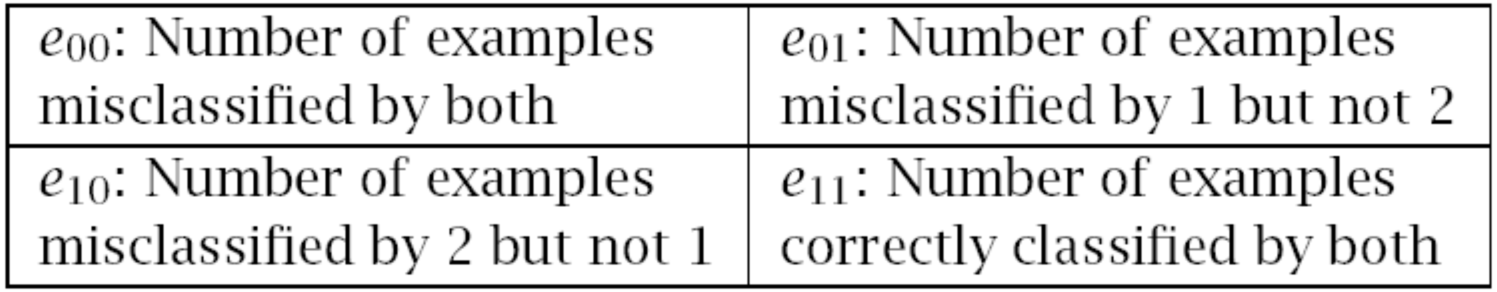
- Multiple training/validation sets, K folds
- x^t_i = 1 if instance *t* misclassified on fold *i*
- Error rate of fold *i*: $$p_i = \frac{\sum_{t=1}^{N} x_i^t}{N}$$
- With *m* and s^2 average and var of p_i , we accept p_0 or less error if

$$\frac{\sqrt{K}\left(m - p_0\right)}{S} \sim t_{K-1}$$

is less than $t_{\alpha,K\text{-}1}$

Comparing Classifiers: H_0: $\mu_0 = \mu_1$ vs. H_1: $\mu_0 \neq \mu_1$

- Single training/validation set: McNemar’s Test



e_{00}: Number of examples misclassified by both	e_{01}: Number of examples misclassified by 1 but not 2
e_{10}: Number of examples misclassified by 2 but not 1	e_{11}: Number of examples correctly classified by both

- Under H_0, we expect $e_{01} = e_{10} = (e_{01} + e_{10})/2$

$$\frac{\left(\left|e_{01} - e_{10}\right| - 1\right)^2}{e_{01} + e_{10}} \sim \chi_1^2$$

Accept if < $X^2_{\alpha,1}$

K-Fold CV Paired *t* Test

- Use *K*-fold cv to get *K* training/validation folds
- p_i^1, p_i^2: Errors of classifiers 1 and 2 on fold *i*
- $p_i = p_i^1 - p_i^2$: Paired difference on fold *i*
- The null hypothesis is whether p_i has mean 0

$$H_0 : \mu = 0 \text{ vs. } H_0 : \mu \neq 0$$

$$m = \frac{\sum_{i=1}^{K} p_i}{K} \qquad s^2 = \frac{\sum_{i=1}^{K} (p_i - m)^2}{K-1}$$

$$\frac{\sqrt{K}(m-0)}{s} = \frac{\sqrt{K} \cdot m}{s} \sim t_{K-1} \text{ Accept if in } \left(-t_{\alpha/2,K-1}, t_{\alpha/2,K-1}\right)$$

5×2 cv Paired *t* Test

- Use 5×2 cv to get 2 folds of 5 tra/val replications (Dietterich, 1998)
- $p_i^{(j)}$: difference btw errors of 1 and 2 on fold *j*=1, 2 of replication *i*=1,...,5

$$\bar{p}_i = \left(p_i^{(1)} + p_i^{(2)}\right)/2 \qquad s_i^2 = \left(p_i^{(1)} - \bar{p}_i\right)^2 + \left(p_i^{(2)} - \bar{p}_i\right)^2$$

$$\frac{p_1^{(1)}}{\sqrt{\sum_{i=1}^{5} s_i^2 / 5}} \sim t_5$$

Two-sided test: Accept H_0: $\mu_0 = \mu_1$ if in $(-t_{\alpha/2,5}, t_{\alpha/2,5})$
One-sided test: Accept H_0: $\mu_0 \leq \mu_1$ if $< t_{\alpha,5}$

5×2 cv Paired *F* Test

$$\frac{\sum_{i=1}^{5}\sum_{j=1}^{2}\left(p_i^{(j)}\right)^2}{2\sum_{i=1}^{5}s_i^2} \sim F_{10,5}$$

Two-sided test: Accept H_0: $\mu_0 = \mu_1$ if < $F_{\alpha,10,5}$

Comparing *L*>2 Algorithms: Analysis of Variance (Anova)

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_L$$

- Errors of *L* algorithms on *K* folds

$$X_{ij} \sim \mathcal{N}\left(\mu_j , \sigma^2\right), j = 1,\ldots,L,\ i = 1,\ldots,K$$

- We construct two estimators to σ^2 .

  One is valid if H_0 is true, the other is always valid.

  We reject H_0 if the two estimators disagree.

If H_0 is true:

$$m_j = \sum_{i=1}^{K} \frac{X_{ij}}{K} \sim \mathcal{N}\left(\mu, \sigma^2 / K\right)$$

$$m = \frac{\sum_{j=1}^{L} m_j}{L} \qquad S^2 = \frac{\sum_j \left(m_j - m\right)^2}{L-1}$$

Thus an estimator of σ^2 is $K \cdot S^2$, namely,

$$\hat{\sigma}^2 = K \sum_{j=1}^{L} \frac{\left(m_j - m\right)^2}{L-1}$$

$$\sum_j \frac{\left(m_j - m\right)^2}{\sigma^2 / K} \sim \chi_{L-1}^2 \quad SSb \equiv K \sum_j \left(m_j - m\right)^2$$

So when H_0 is true, we have

$$\frac{SSb}{\sigma^2} \sim \chi_{L-1}^2$$

Regardless of H_0 our second estimator to σ^2 is the average of group variances S_j^2 :

$$S_j^2 = \frac{\sum_{i=1}^{K}\left(X_{ij} - m_j\right)^2}{K-1} \quad \hat{\sigma}^2 = \sum_{j=1}^{L}\frac{S_j^2}{L} = \sum_j\sum_i \frac{\left(X_{ij} - m_j\right)^2}{L(K-1)}$$

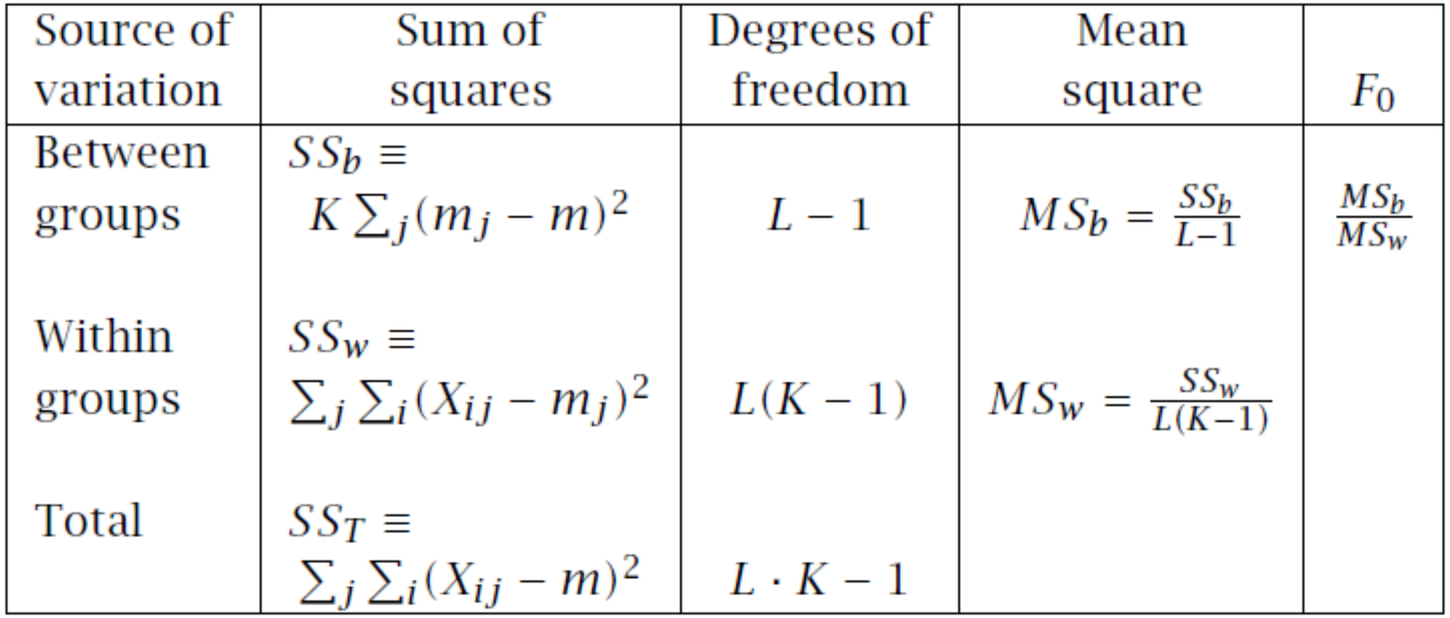
$$SSw \equiv \sum_j\sum_i\left(X_{ij} - m_j\right)^2$$

$$(K-1)\frac{S_j^2}{\sigma^2} \sim \chi_{K-1}^2 \quad \frac{SSw}{\sigma^2} \sim \chi_{L(K-1)}^2$$

$$\left(\frac{SSb/\sigma^2}{L-1}\right)\Big/\left(\frac{SSw/\sigma^2}{L(K-1)}\right) = \frac{SSb/(L-1)}{SSw/(L(K-1))} \sim F_{L-1,L(K-1)}$$

$H_0 : \mu_1 = \mu_2 = \cdots = \mu_L$ if $< F_{\alpha,L-1,L(K-1)}$

ANOVA table



Source of variation	Sum of squares	Degrees of freedom	Mean square	F_0
Between groups	$SS_b \equiv K\sum_j (m_j - m)^2$	$L - 1$	$MS_b = \frac{SS_b}{L-1}$	$\frac{MS_b}{MS_w}$
Within groups	$SS_w \equiv \sum_j \sum_i (X_{ij} - m_j)^2$	$L(K - 1)$	$MS_w = \frac{SS_w}{L(K-1)}$	
Total	$SS_T \equiv \sum_j \sum_i (X_{ij} - m)^2$	$L \cdot K - 1$		

If ANOVA rejects, we do pairwise posthoc tests

$$H_0 : \mu_i = \mu_j \text{ vs } H_1 : \mu_i \neq \mu_j$$

$$t = \frac{m_i - m_j}{\sqrt{2}\sigma_w} \sim t_{L(K-1)}$$

IMPORTANT

You need to test the normality assumption prior to conducting significance tests. If the normality assumption fails, you need to pick non-parametric significance tests.

Comparison over Multiple Datasets

- Comparing two algorithms:

  Sign test: Count how many times *A* beats *B* over *N* datasets, and check if this could have been by chance if A and B did have the same error rate

- Comparing multiple algorithms

  Kruskal-Wallis test: Calculate the average rank of all algorithms on N datasets, and check if these could have been by chance if they all had equal error

  If KW rejects, we do pairwise posthoc tests to find which ones have significant rank difference