Statistics 101

Today's Topics

→ Various distributions of random variables and their display

Characterizing distributions; central tendency and dispersion

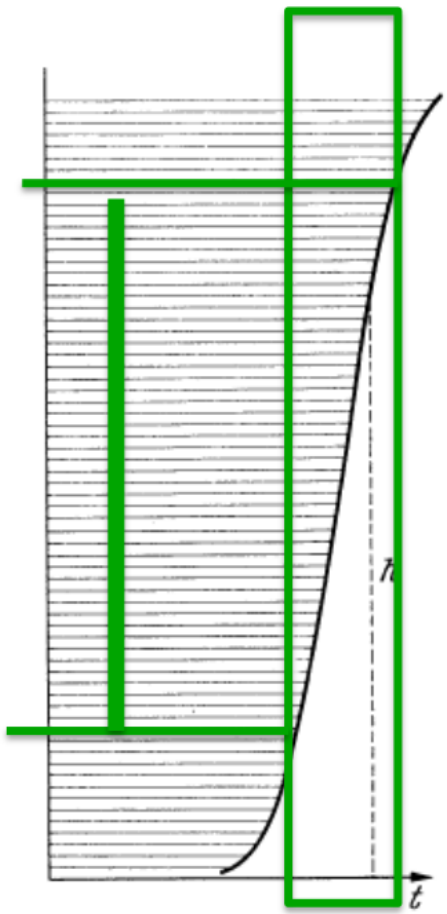
Some theorems on distributions (with/without models)

Standard error of the empirical mean and empirical variance

Error propagation in regression for measurements with precision

Hypothesis testing

Example: The Quetelet Curve



"I still remember vividly how my father took me one day as a boy to the outskirts of the city, where the willows stood on the bank and had me pick a hundred willow leaves at random. After throwing out those with damaged tips, 89 were left which we took home. We arranged them very carefully according to decreasing size like soldiers in rank and file.

Then my father drew a curved line through the tips and said, "This is the curve of Quetelet. From it you see how the mediocre always form the large majority and only a few stand out or are left behind.""

B.L. van der Waerden, Mathematical Statistics, Springer 1968

Two Views of Statistics

Descriptive statistics

Statistical methods to describe data in the form of diagrams, tables, or individual parameters

Inferential statistics

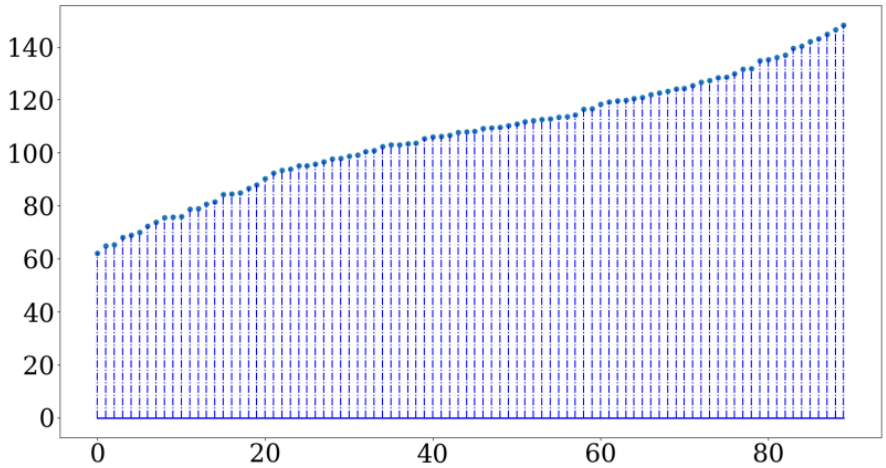
Inferring the validity of hypotheses from given empirical data based on qualitative analysis techniques

Example: Raw Data

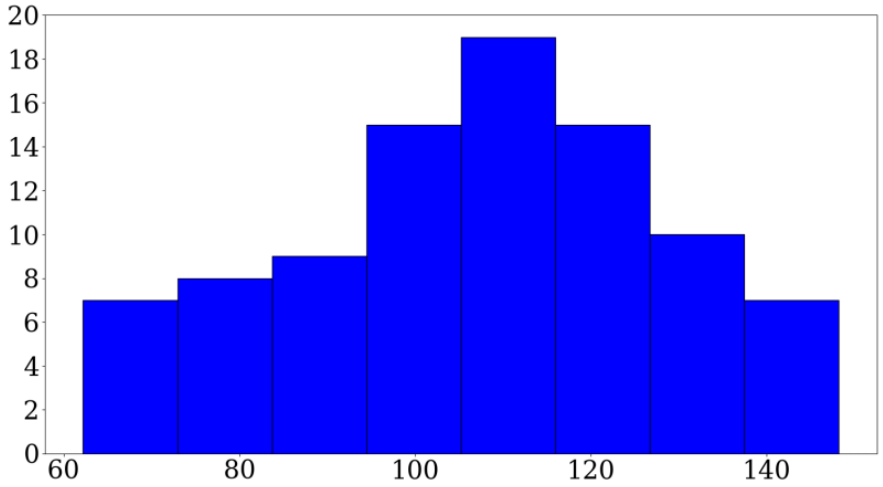
131.8	106.7	116.4	84.3	118.5	93.4	65.3	113.8	140.3
119.2	129.9	75.7	105.4	123.4	64.9	80.7	124.2	110.9
86.7	112.7	96.7	110.2	135.2	134.7	146.5	144.8	113.4
128.6	142.0	106.0	98.0	148.2	106.2	122.7	70.0	73.9
78.8	103.4	112.9	126.6	119.9	62.2	116.6	84.6	101.0
68.1	95.9	119.7	122.0	127.3	109.3	95.1	103.1	92.4
103.0	90.2	136.1	109.6	99.2	76.1	93.9	81.5	100.4
114.3	125.5	121.0	137.0	107.7	69.0	79.0	111.7	98.8
124.3	84.9	108.1	128.5	87.9	102.4	103.7	131.7	139.4
108.0	109.4	97.8	112.2	75.6	143.1	72.4	120.6	95.2

```
data = np.array([131.8, 106.7, 116.4, 84.3, 118.5, 93.4, 65.3, 113.8, 140.3, 119.2, 129.9,
      75.7, 105.4, 123.4, 64.9, 80.7, 124.2, 110.9, 86.7, 112.7, 96.7, 110.2, 135.2, 134.7,
      146.5, 144.8, 113.4, 128.6, 142.0, 106.0, 98.0, 148.2, 106.2, 122.7, 70.0, 73.9, 78.8,
      103.4, 112.9, 126.6, 119.9, 62.2, 116.6, 84.6, 101.0, 68.1, 95.9, 119.7, 122.0, 127.3,
      109.3, 95.1, 103.1, 92.4, 103.0, 90.2, 136.1, 109.6, 99.2, 76.1, 93.9, 81.5, 100.4,
      114.3, 125.5, 121.0, 137.0, 107.7, 69.0, 79.0, 111.7, 98.8, 124.3, 84.9, 108.1, 128.5,
      87.9, 102.4, 103.7, 131.7, 139.4, 108.0, 109.4, 97.8, 112.2, 75.6, 143.1, 72.4, 120.6,
      95.2])
```

Example: Different Plots of the Data (1/2)



Stem plot

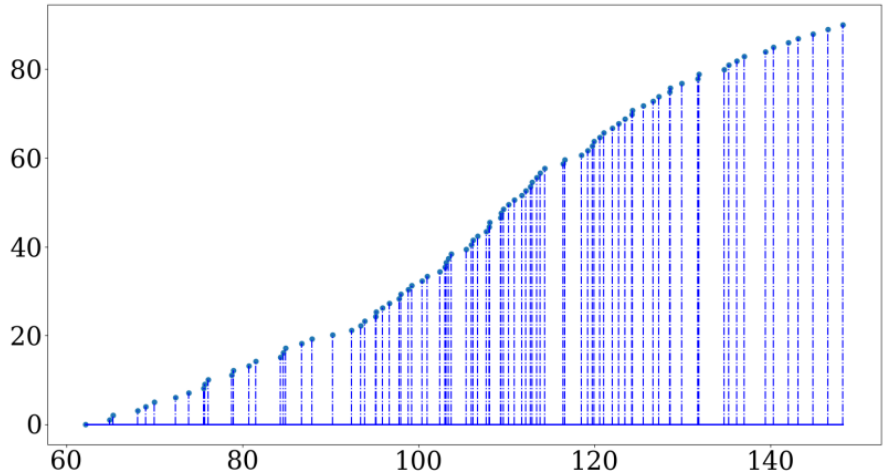


Histogram plot

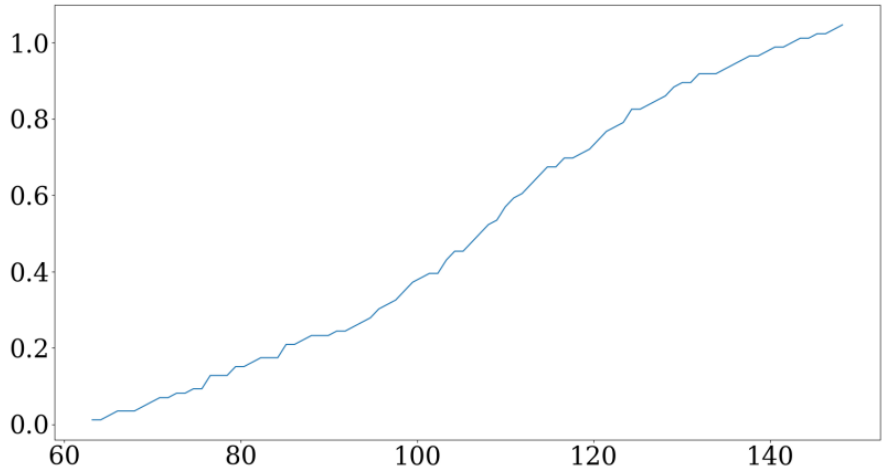
```
markerline, stemlines, baseline = plt.stem(np
      .sort(data), '-.', color='b')
plt.setp(baseline, 'color', 'b', 'linewidth',
       2)
plt.setp(stemlines, 'color', 'b')
plt.xticks(fontsize=32)
plt.yticks(fontsize=32)
plt.show()
```

```
plt.hist(data, bins=8, color='b', edgecolor='
      k')
plt.xticks(fontsize=32)
plt.yticks([0, 2, 4, 6, 8, 10, 12, 14, 16,
      18, 20], fontsize=32)
plt.show()
```

Example: Different Plots of the Data (2/2)



Quantile plot

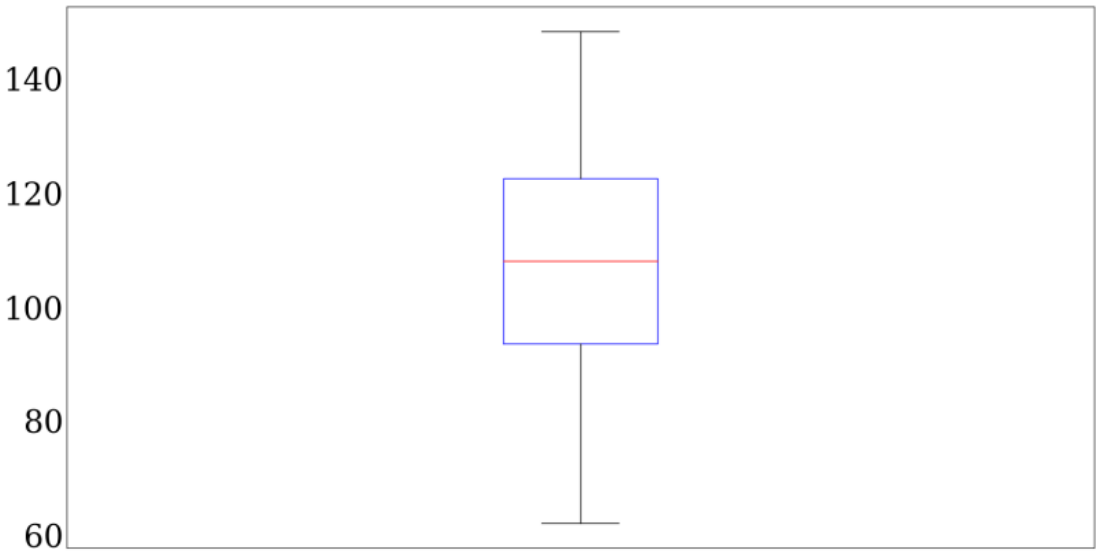


Empirical CDF plot

```
markerline, stemlines, baseline = plt.stem(np
      .sort(data), np.linspace(0, len(data),
      len(data)), '-.', color='b')
plt.setp(baseline, 'color', 'b', 'linewidth',
       2)
plt.setp(stemlines, 'color', 'b')
plt.xticks(fontsize=32)
plt.yticks(fontsize=32)
plt.show()
```

```
num_bins = len(data)
counts, bin_edges = np.histogram(data, bins=
      num_bins, normed=True)
cdf = np.cumsum(counts)
plt.plot(bin_edges[1:], cdf)
plt.xticks(fontsize=32)
plt.yticks(fontsize=32)
plt.show()
```

Boxplot



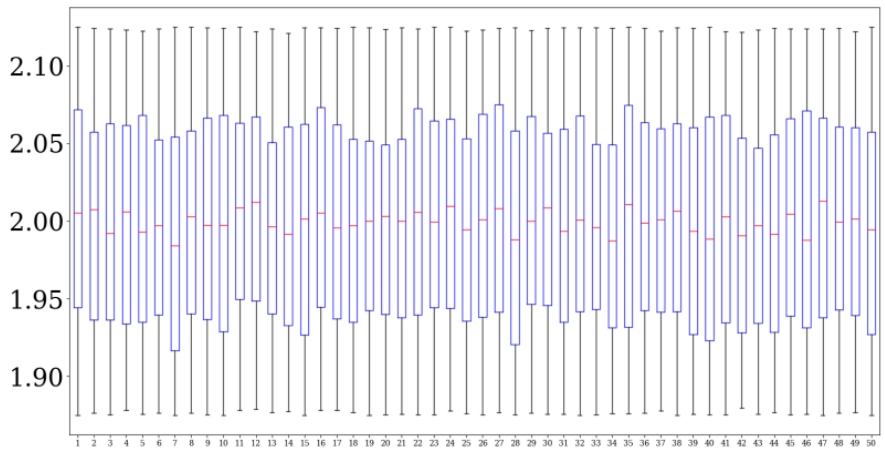
A boxplot shows

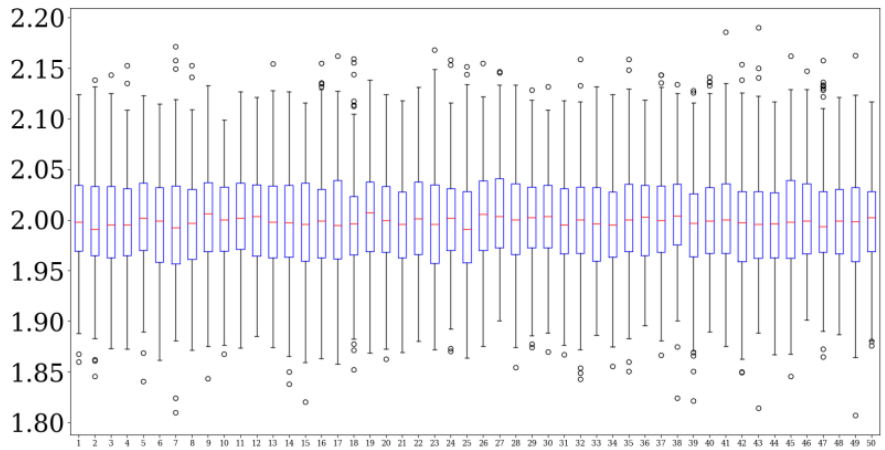
→ The median (red) (50% quantile)
→ The lower 25% quantile (lower box line lbl)
→ Upper 75% quantile (upper box line ubl)
→ Whiskers $==$ median $\pm 1.5 \cdot (ubl - lbl)$
→ Outliers (data beyond wiskers)
→ 99.3% coverage of the normal distribution

Boxplots are Good for Overview of Spaghetti Data

```
import numpy as np
import matplotlib.pyplot as plt

spaghetti_50 = (np.random.rand(250,50) -0.5 )
       * 0.25 + 2

ax = plt.gca()
bp = ax.boxplot(spaghetti_50)
plt.setp(bp['boxes'], color='b')
plt.setp(bp['whiskers'], color='k')
plt.setp(bp['medians'], color='r')
plt.xticks(np.linspace(1, 50, 50))
plt.yticks(fontsize=32)
plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt

spaghetti_50n = np.random.normal
      (2,0.05,(250,50))

ax = plt.gca()
bp = ax.boxplot(spaghetti_50n)
plt.setp(bp['boxes'], color='b')
plt.setp(bp['whiskers'], color='k')
plt.setp(bp['medians'], color='r')
plt.xticks(np.linspace(1, 50, 50))
plt.yticks(fontsize=32)
plt.show()
```





Basic Definitions and Facts in Statistics (1/3)

Random variable

A random variable X can be viewed as the name of a (real-valued) experiment with a probabilistic outcome, such that its value is the outcome of the experiment

Probability distribution

The probability distribution of a random variable X specifies the probability $P(X = x_i)$ that X will take on the value x_i for each possible value x_i

Expected value (mean)

The expected value of a random variable Y is defined as $E[X] = \sum_i x_i P(X = x_i)$. $E[X]$ is commonly denoted as μ

Variance

The variance of a random variable is defined as $Var(X) = E[(X - E[X])^2]$. The variance characterizes the width or dispersion of the distribution about its mean

Basic Definitions and Facts in Statistics (2/3)

Standard deviation

The standard deviation of X is defined as $\sigma = Var(X)$

Binomial distribution

The Binomial distribution gives the probability of observing r heads in a series of n independent coin tosses, given that the probability of heads in a single toss is p

Normal (Gaussian) distribution

The normal distribution is a bell-shaped probability distribution that covers many natural phenomena

Central Limit Theorem

The Central Limit Theorem states that the mean value of a large number of independent, identically distributed random variables approximately follows a normal distribution

Basic Definitions and Facts in Statistics (3/3)

Estimator

An estimator is a random variable X used to estimate some parameter p of an underlying population

Estimation bias

The estimation bias of X as an estimator of p is the quantity $E[X] - p$. An unbiased estimator is one for which the bias is zero

$N\%$ confidence interval

An $N\%$ confidence interval estimate of a parameter p is an interval that includes p with probability $N\%$

Random Variables

Probability measure

A probability measure P is a map from an algebra of sets $\mathcal{A}$

$$P : \mathcal{A} \mapsto \mathbb{R}$$

namely $A \mapsto P(A)$, which obeys Kolmogorov's axioms:

1. $P(A) \geq 0$
2. $P(\Omega) = 1$
3. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

Random variable

A random variable is a real-valued function

$$X : \Omega \mapsto \mathbb{R}$$

namely $\omega \mapsto X(\omega)$ where ω is the result of a random experiment (such as for example the measured runtime of an algorithm on a CPU with cache). X needs to be measurable

Cumulative Distribution Function (CDF) and Quantile Function

Cumulative distribution function (CDF)

The cumulative distribution function is defined as

$$F_X : \mathbb{R} \mapsto [0,1]$$
$$x \mapsto F_X : P(\{\omega | X(\omega) \leq x\})$$

Quantile function

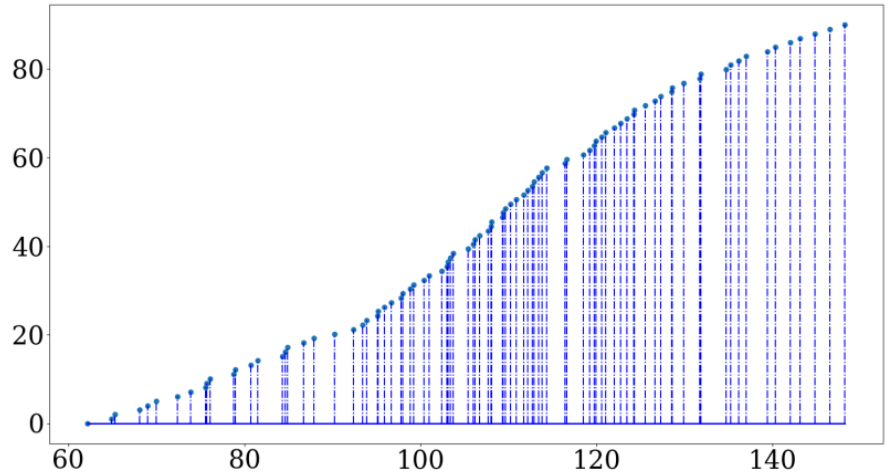
If F_X is a CDF, then the function

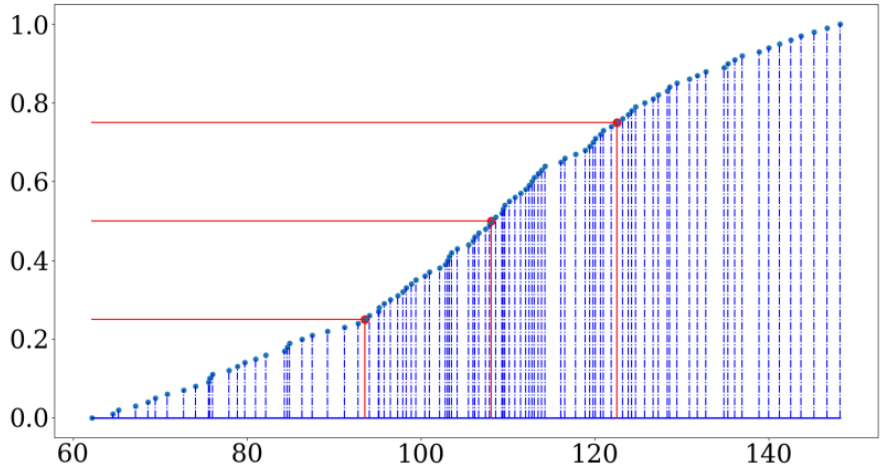
$$Q_X : [0,1] \mapsto \mathbb{R}$$
$$p \mapsto Q_X(p) = \min\{x | F_X(x) \geq p\}$$

is called the quantile function of F_X. Q_X is the inverse of F_X in the following sense:

$$F_X(Q_X(p)) \geq p \text{ for } p \in [0,1] \text{ and}$$
$$Q_X(F_X(p)) \leq x \text{ for } x \in \mathbb{R}$$

Quantile Function == Inverse CDF






```
markerline, stemlines, baseline = plt.stem(np
      .sort(data), np.linspace(0, len(data),
      len(data)), '-.', color='b')
plt.setp(baseline, 'color', 'b', 'linewidth',
       2)
plt.setp(stemlines, 'color', 'b')
plt.xticks(fontsize=32)
plt.yticks(fontsize=32)
plt.show()
```

```
p = np.linspace(0,100,101)
percentiles = matplotlib.mlab.prctile(data, p
      =p)
markerline, stemlines, baseline = plt.stem(
      percentiles, np.linspace(0,1,101), '-.'
      , color='b')
quantiles = np.array([percentiles[p[25]],
      percentiles[p[50]], percentiles[p
      [75]]])

plt.plot(quantiles, np.array([0.25, 0.5,
      0.75]), 'ro')
plt.setp(baseline, 'color', 'b', 'linewidth',
       2)
plt.setp(stemlines, 'color', 'b')
plt.xticks(fontsize=32)
plt.yticks(fontsize=32)
plt.show()
```

How to Create Histograms

By increasing the size of the examined collective, smaller category widths can be selected

The larger the variation of the measured values (the difference between the largest and the smallest value), the wider the categories may be

According to a a rule of thumb by Sturges (1926), the number of categories m should be determined according to the relation $m \approx 1 + 3.32 \log(n)$, where n is the sample size

The maximum number of categories should not exceed 20 for the sake of clarity

All categories should in general have the same width

Various distributions of random variables and their display

→ Characterizing distributions; central tendency and dispersion

Some theorems on distributions (with/without models)

Standard error of the empirical mean and empirical variance

Error propagation in regression for measurements with precision

Hypothesis testing

Plots Versus Summarising Statistics

While the plot of a distribution stresses the WHOLE distribution (the whole collection of data is represented), statistical parameters should stress special global summarising features of the underlying distribution

In other words, they should represent "typical" attributes, such as the general tendency or variability, i.e. the dispersion of the distribution

Example: Summarising Statistics vs. Data Plots (1/2)

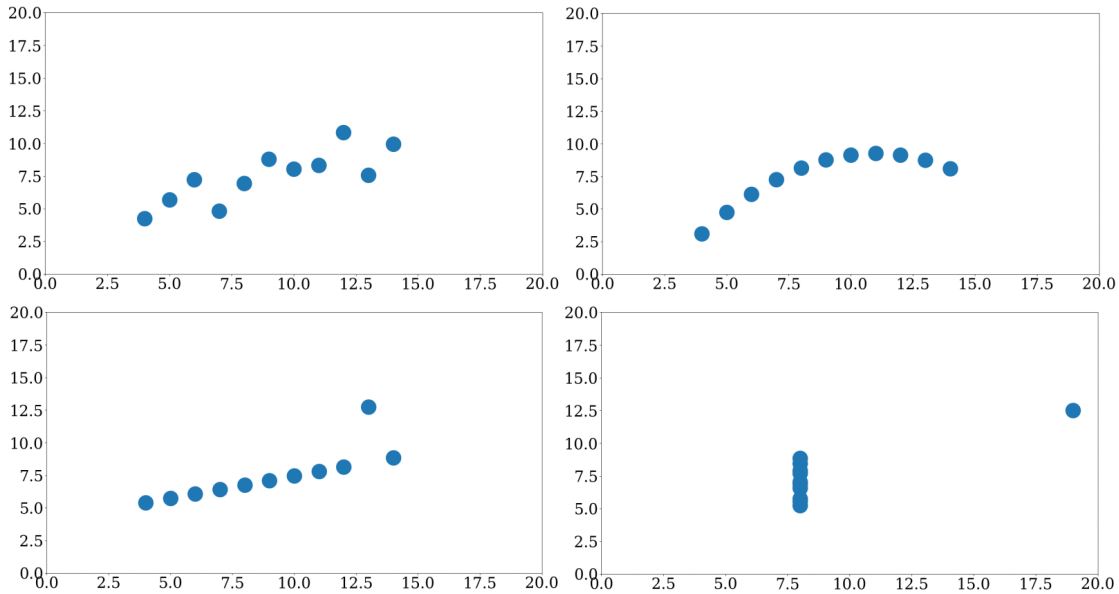
X	Y	X	Y	X	Y	X	Y
10.0000	8.0400	10.0000	9.1400	10.0000	7.4600	8.0000	6.5800
8.0000	6.9500	8.0000	8.1400	8.0000	6.7700	8.0000	5.7600
13.0000	7.5800	13.0000	8.7400	13.0000	12.7400	8.0000	7.7100
9.0000	8.8100	9.0000	8.7700	9.0000	7.1100	8.0000	8.8400
11.0000	8.3300	11.0000	9.2600	11.0000	7.8100	8.0000	8.4700
14.0000	9.9600	14.0000	8.1000	14.0000	8.8400	8.0000	7.0400
6.0000	7.2400	6.0000	6.1300	6.0000	6.0800	8.0000	5.2500
4.0000	4.2600	4.0000	3.1000	4.0000	5.3900	19.0000	12.5000
12.0000	10.8400	12.0000	9.1300	12.0000	8.1500	8.0000	5.5600
7.0000	4.8200	7.0000	7.2600	7.0000	6.4200	8.0000	7.9100
5.0000	5.6800	5.0000	4.7400	5.0000	5.7300	8.0000	6.8900

$N = 11$
Mean of $X = 9.0$
Mean of $Y = 7.5$
Equation of regression line: $Y = 3 + \frac{1}{2}X$
Standard error of slope estimate $= 0.118$

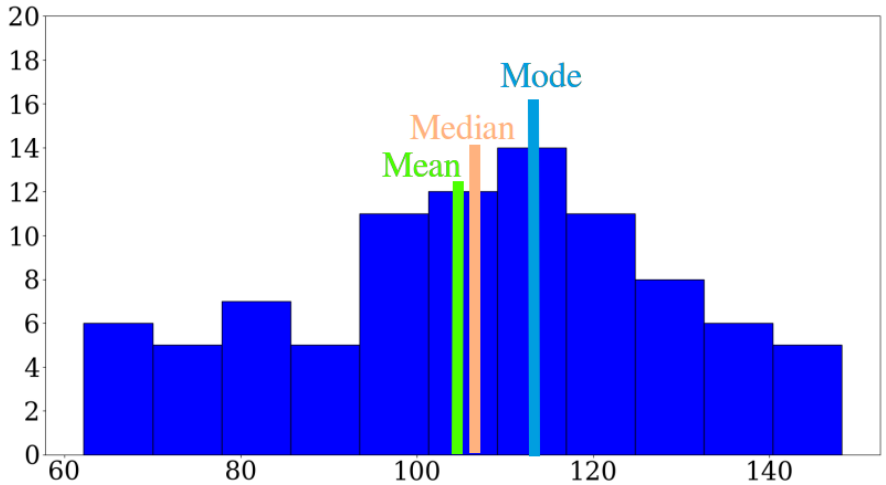
$t = 4.24$
Sum of squares $X - X^2 = 110.0$
Regression sum of squares $= 27.50$
Residual sum of squares of $Y = 13.75$
Correlation coefficient $= 0.82$

Example: Summarising Statistics vs. Data Plots (2/2)

```
data = np.reshape(data, (11, 8))
plt.scatter(data[:,0], data[:,1])
plt.xlim([0,20])
plt.ylim([0,20])
plt.show()

plt.scatter(data[:,2], data[:,3])
plt.xlim([0,20])
plt.ylim([0,20])
plt.show()

plt.scatter(data[:,4], data[:,5])
plt.xlim([0,20])
plt.ylim([0,20])
plt.show()

plt.scatter(data[:,6], data[:,7])
plt.xlim([0,20])
plt.ylim([0,20])
plt.show()
```



Which Single Value is the Most Characteristic of a Distribution?




```
counts, values = np.hist(data, bins=11)
mode = values[np.where(counts == np.max(
      counts))[0][0]]
```

```
median = np.median(data)
```

```
mean = np.mean(data)
```

Mode

The most frequent value of a distribution (note: a **unimodal** distribution has only one peak; **multimodal** distributions have multiple peaks)

Median

50% of the data are below this value

Empirical mean value

Closest to the data on average

Mode and Median

Mode of a frequency distribution

	Measurement x	Frequency $f(x)$
	11	2
	12	8
	13	18
	14	17
	15	22
Mode	16	28
	17	21
	18	11
	19	3

Median of grouped data

Error count k	Frequency f	Cumulative frequency
$1-20$	3	3
$21-40$	16	19
$41-60$	12	31
$61-80$	7	38
$81-100$	5	43
$101-120$	4	47
$121-140$	3	50

$$median = u + \frac{\frac{n}{2} - F}{f_{MD}} Kb$$

u - lower limit of the category containing a hit
Kb - interval width
F - count of cases in the hit category, but below the median
f_{MD} - case count in the hit category

Empirical Mean $\bar{x}$

The empirical mean lies optimally "in the middle", minimising the sum of distances; in addition, the mean equilibrates the difference to all elements

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{x})^2$$

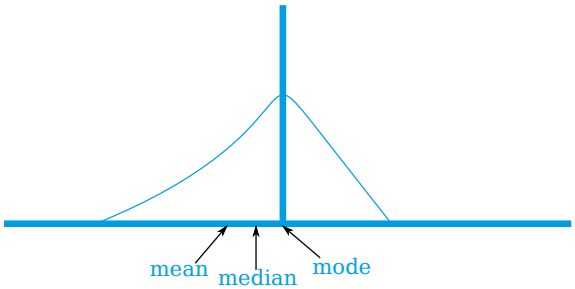
$$\begin{aligned}\sum (x_i - \bar{x}) &= \sum x_i - \sum \bar{x} \\ &= \sum x_i - n\bar{x} = \sum x_i - n\frac{1}{n}\sum x_i = 0\end{aligned}$$

The mean is also minimal for the sum of squared differences

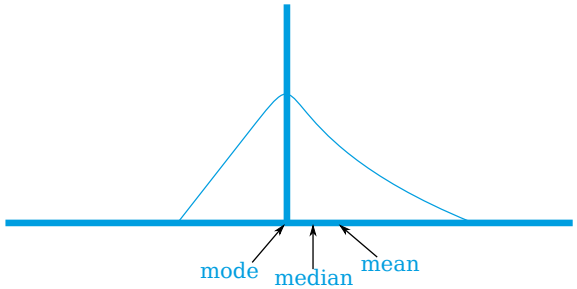
$$\min_c \sum (x_i - c)^2 = \bar{x}$$

Comparing the Mode, Median, and Empirical Mean

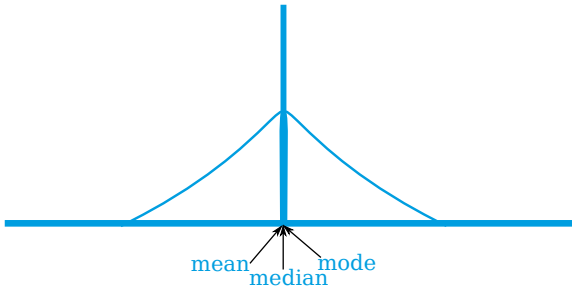
In case of a symmetric distribution, the mean, median, and mode are the same; for biased distributions (skewed in one direction), they are not



Left-skewed distribution



Right-skewed distribution



Symmetric distribution

Dispersion (1/2)

Mean average deviation (AD)

$$AD = \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n}$$

Grade x	$\lvert x_i - \bar{x} \rvert$
3.3	0.8
1.7	0.8
2.0	0.5
4.0	1.5
1.3	1.2
2.0	0.5
3.0	0.5
2.7	0.2
3.7	1.2
2.3	0.2
1.7	0.8
2.3	0.2
$\sum x_i = 30$	$\sum \lvert x_i - \bar{x} \rvert = 8.4$

$$\bar{x} = 2.5$$

$$AD = \frac{8.4}{12} = 0.7$$

Dispersion (2/2)

Empirical variance

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}$$

Empirical standard deviation

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}}$$

Remark: for small n, it is better to divide by $n-1$ rather than by n (unbiased estimator)!

Grade x	$x_i - \bar{x}$	$(x - \bar{x})^2$
3.3	0.8	0.64
1.7	−0.8	0.64
2.0	−0.5	0.25
4.0	1.5	2.25
1.3	−1.2	1.44
2.0	−0.5	0.25
3.0	0.5	0.25
2.7	0.2	0.04
3.7	1.2	1.44
2.3	−0.2	0.04
1.7	−0.8	0.64
2.3	−0.2	0.04
$\sum x_i = 30$	$\sum x_i - \bar{x} = 0$	$\sum (x_i - \bar{x})^2 = 7.92$

$$\bar{x} = 2.5$$

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n} = \frac{7.92}{12} = 0.66$$

$$s = \sqrt{0.66} = 0.81$$

Various distributions of random variables and their display

Characterizing distributions; central tendency and dispersion

→ Some theorems on distributions (with/without models)

Standard error of the empirical mean and empirical variance

Error propagation in regression for measurements with precision

Hypothesis testing

Chebyshev's Inequality

If X is a random variable with $E[X] = \mu$ and standard deviation σ, $k > 0$, it holds that

$$P(|X - \mu| \leq k\sigma) \geq \left(1 - \frac{1}{k^2}\right)$$

This determines a lower bound of the data part that falls within k numbers of standard deviations from the mean

Taking complements (focusing on data away from the mean, i.e. outliers), we have

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

Example: Hospital Patients

Let X be the number of days a patient spends in a hospital; the expected value of X is $\bar{x} = 10$ and the standard deviation is $\sigma = 4$. How likely is it that a patient stays more than five and less than 15 days in the hospital?

Example: Hospital Patients

Let X be the number of days a patient spends in a hospital; the expected value of X is $\bar{x} = 10$ and the standard deviation is $\sigma = 4$. How likely is it that a patient stays more than five and less than 15 days in the hospital?

Solution

$$P(5 < X < 15) = P(|X < 10| < 5) \geq 1 - \frac{4 \cdot 4}{5 \cdot 5} = \frac{9}{25}$$

Estimating X Within Two or Three σ Bounds

Chebyshev on P

$$P(\mu - 2\sigma < X < \mu + 2\sigma) \geq \frac{3}{4} = 0.75$$
$$P(\mu - 3\sigma < X < \mu + 3\sigma) \geq \frac{8}{9} = 0.88$$

If we have additional knowledge that P is normal, we get

Chebyshev on P

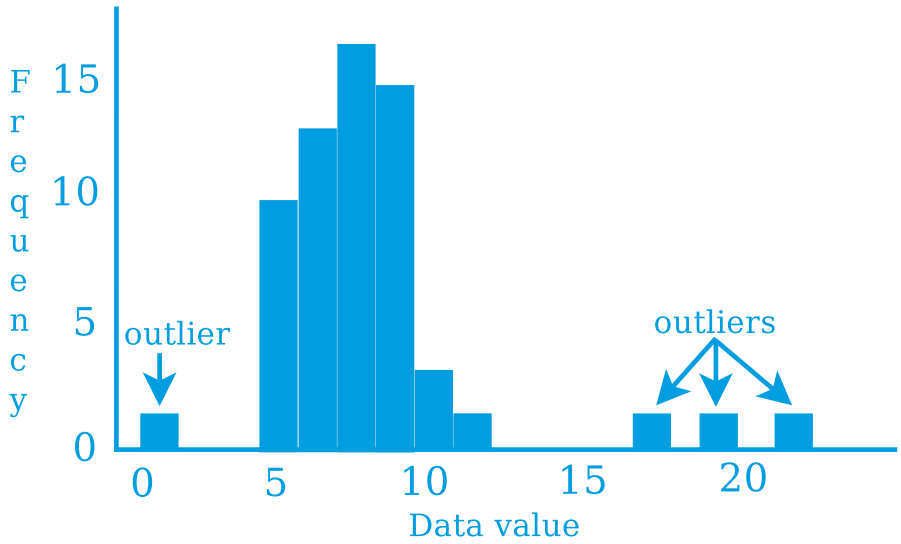
$$P(\mu - 2\sigma < X < \mu + 2\sigma) = 2\Phi(2) - 1 = 0.9544$$
$$P(\mu - 3\sigma < X < \mu + 3\sigma) = 2\Phi(3) - 1 = 0.9974$$

Practical Issues

In practice, we have to use empirical values for the mean and variance, which are affected by outliers

$$\mu \approx \bar{x} = \frac{1}{n} \sum x_i$$

$$\sigma^2 \approx s_x^2 = \frac{1}{n-1} \sum (x_i - \bar{x})^2$$



Chebyshev's Outlier Algorithm

```
def remove_outliers(data, pp1, pp2):
    '''
    Based on "Data Outlier Detection using the Chebyshev Theorem",
    Brett G. Amidan, Thomas A. Ferryman, and Scott K. Cooley

    Keyword arguments:
    data -- A numpy array of discrete or continuous data
    pp1 -- likelihood of expected outliers (e.g. 0.1, 0.05 , 0.01)
    pp2 -- final likelihood of real outliers (e.g. 0.01, 0.001 , 0.0001)
    '''
    mu1 = np.mean(data)
    sigma1 = np.std(data)
    k = 1./ np.sqrt(pp1)
    odv1u = mu1 + k * sigma1
    odv1l = mu1 - k * sigma1
    new_data = data[np.where(data <= odv1u)[0]]
    new_data = new_data[np.where(new_data >= odv1l)[0]]

    mu2 = np.mean(new_data)
    sigma2 = np.std(new_data)
    k = 1./ np.sqrt(pp2)
    odv2u = mu2 + k * sigma2
    odv2l = mu2 - k * sigma2
    final_data = new_data[np.where(new_data <= odv2u)[0]]
    final_data = new_data[np.where(final_data >= odv2l)[0]]
    return final_data
```

Various distributions of random variables and their display

Characterizing distributions; central tendency and dispersion

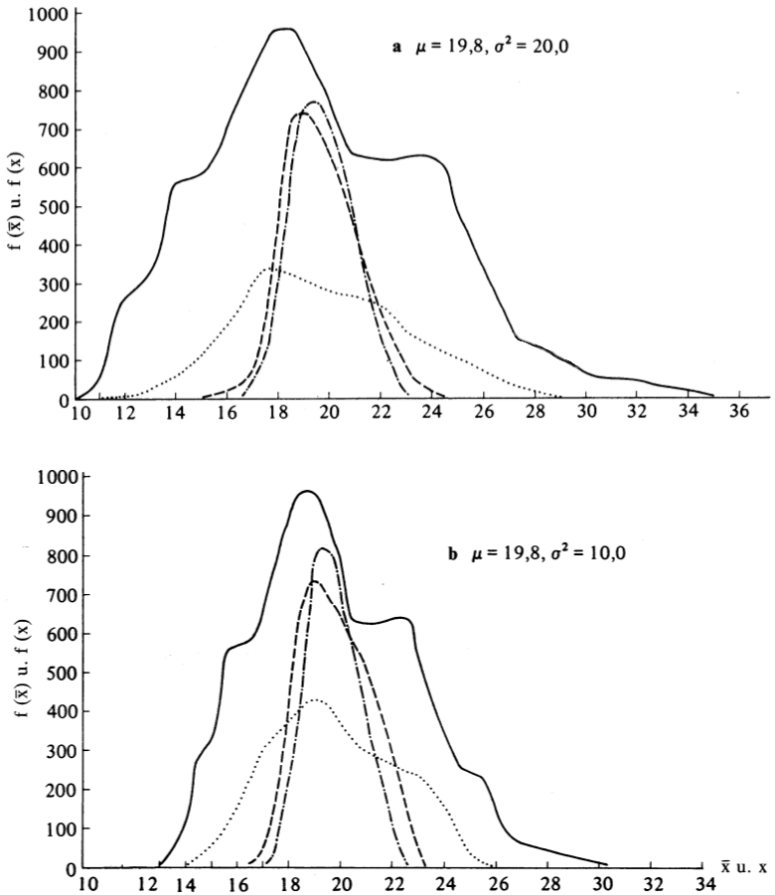
Some theorems on distributions (with/without models)

→ Standard error of the empirical mean and empirical variance

Error propagation in regression for measurements with precision

Hypothesis testing

Examples of Standard Error



Solid line: population; dotted line: mean distribution of 200 samples ($n = 2$); dashed line: mean distribution of 200 samples ($n = 10$); dash-dotted line: mean distribution of 200 samples ($n = 20$)

$\bar{x}_{\bar{x}}$	$\sigma^2_{\bar{x}}$	$\sigma_{\bar{x}}$	$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$
19.9	9.7	3.1	3.2
19.8	1.8	1.3	1.4
19.8	1.0	1.0	1.0
		empirical	theoretical

$\bar{x}_{\bar{x}}$	$\sigma^2_{\bar{x}}$	$\sigma_{\bar{x}}$	$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$
19.7	5.1	2.3	2.2
19.8	1.2	1.1	1.0
19.8	0.4	0.6	0.7
		empirical	theoretical

The standard error is dependent on the population standard deviation σ and the sample size n

Standard Errors of the Mean and Standard Deviation

$\sigma_{\bar{x}}$ is defined as the *standard error of the mean*, which is the standard deviation of equally sized samples taken from one common population

If the σ of the population is known, we have

$$\sigma_{\bar{x}} = \sqrt{\frac{\sigma^2}{n}} = \frac{\sigma}{\sqrt{n}}$$

In general, σ itself must be estimated

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}$$

$$\Longrightarrow \hat{\sigma}_{\bar{x}} = \sqrt{\frac{\hat{\sigma}^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n(n-1)}}$$

The smaller $\hat{\sigma}_{\bar{x}}$ is, the better μ has been estimated

Standard error of the standard deviation

$$\hat{\sigma}_s = \sqrt{\frac{\hat{\sigma}^2}{2n}}$$

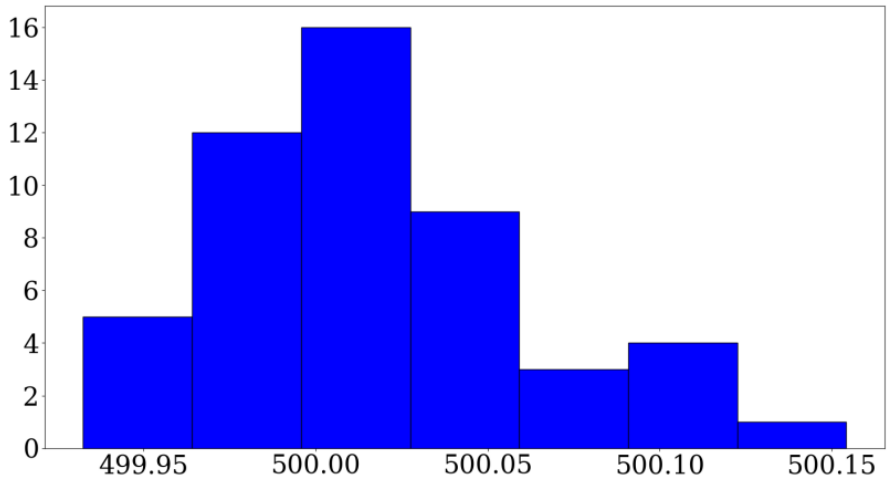
Example: Spaghetti Factories (1/2)

Which spaghetti factory makes better $500g$ packages?

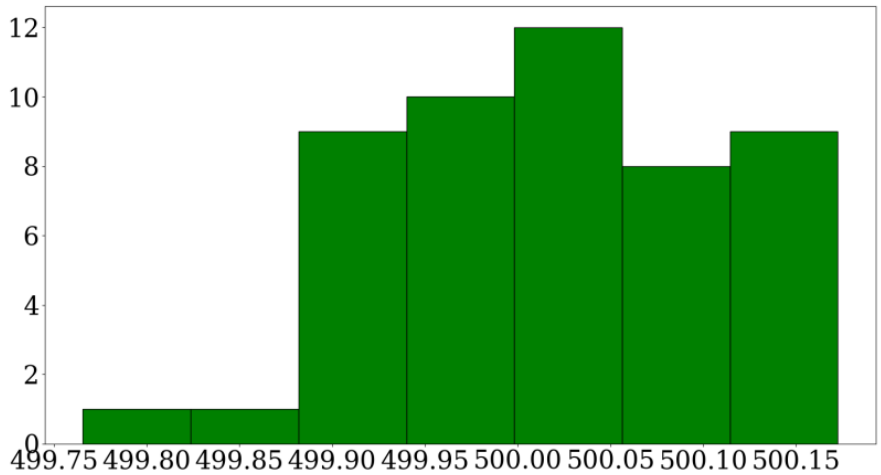
```
import numpy as np
import matplotlib.pyplot as plt

def std_error_of_mean(dist, sample_size, repeat_count):
    '''
    Keyword arguments:
    dist -- A one-dimensional numpy array
    sample_size -- each sample drawn from dist has this size (e.g. 200)
    repeat_count -- how many samples are used to calculate the mean (e.g 50)
    '''
    box_num = len(dist)
    indices = np.array(np.linspace(0,box_num-1,box_num), dtype=int)
    r_dist = np.zeros(repeat_count)
    for k in xrange(0,repeat_count):
        random_indices = np.random.permutation(indices)[:sample_size]
        current_sum = np.sum(dist[random_indices])
        r_dist[k] = current_sum / sample_size
    return r_dist

spaghetti_10000n = np.random.normal(2,0.05,(250,10000))
spaghetti_10000 = (np.random.rand(250,10000) -0.5 ) * 0.25 + 2
box_10000n = np.sum(spaghetti_10000n, axis=0)
box_10000 = np.sum(spaghetti_10000, axis=0)
normal_samples = std_error_of_mean(box_10000n, 200, 50)
samples = std_error_of_mean(box_10000, 200, 50)
plt.hist(normal_samples, bins=int(np.sqrt(len(normal_samples))), color='b', edgecolor='k')
plt.show()
plt.hist(samples, bins=int(np.sqrt(len(samples))), color='g', edgecolor='k')
plt.show()
```

Example: Spaghetti Factories (1/2)

Which spaghetti factory makes better $500g$ packages?



Normally distributed data



Uniformly distributed data

Various distributions of random variables and their display

Characterizing distributions; central tendency and dispersion

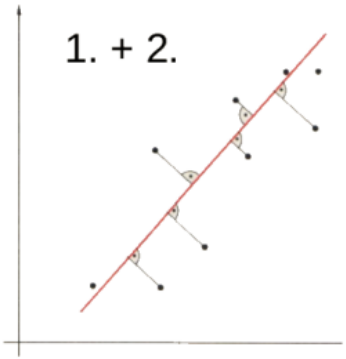
Some theorems on distributions (with/without models)

Standard error of the empirical mean and empirical variance

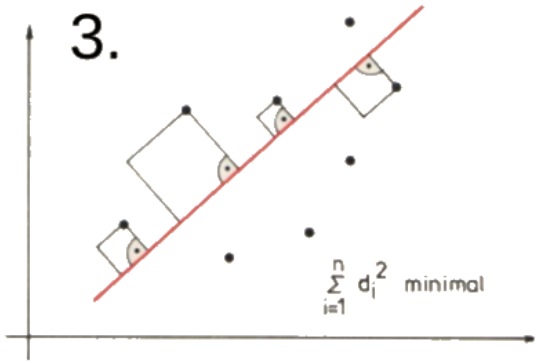
→ Error propagation in regression for measurements with precision

Hypothesis testing

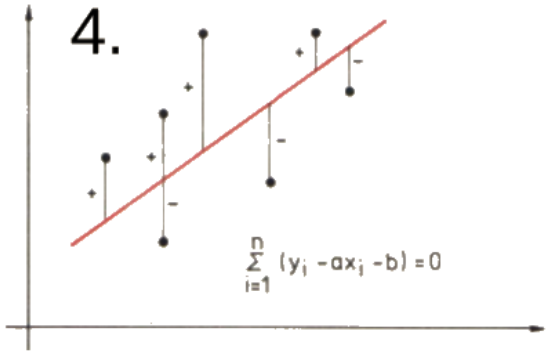
Regression: Distance Metrics



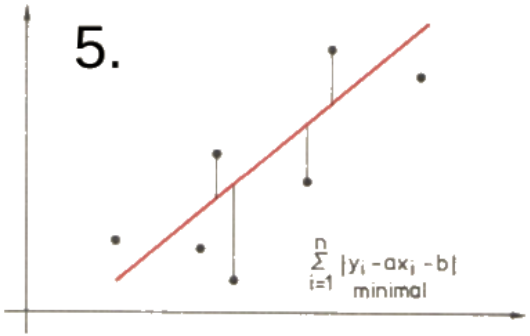
1. Sum of distances above = sum of distances below
2. Sum of orthogonal distances is minimal



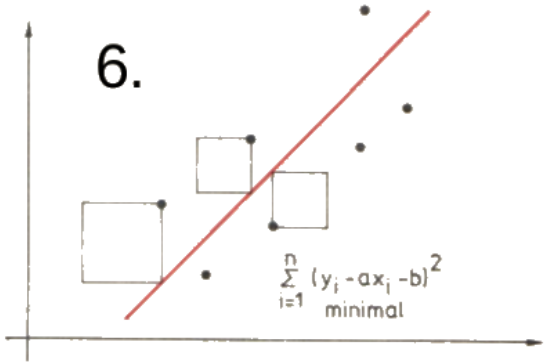
3. Sum of squares is minimal



4. Sum of straight line displacements is zero



5. Sum of absolute values of line displacements is minimal



6. Sum of squares of straight line displacements is minimal

Sum of Squares: Parameter Derivation

$$F(a,b) = \sum_i (y_i - (ax_i + b))^2 = \text{minimum}$$

$$\frac{\partial}{\partial b} F(a,b) = -2 \sum_i (y_i - ax_i - b) = 0$$

$$\implies nb = \sum_i y_i - a \sum_i x_i$$

$$\implies b = \frac{1}{n} \sum_i y_i - a \frac{1}{n} \sum_i x_i = E[Y] - aE[X]$$

$$\begin{aligned}
\frac{\partial}{\partial a} F(a,b) &= \frac{\partial}{\partial a} \sum_i (y_i - (ax_i + E[Y] - aE[X]))^2 \\
&= \frac{\partial}{\partial a} \sum_i [(y_i - E[Y]) - a(x_i - E[X])]^2 \\
&= \frac{\partial}{\partial a} \sum_i (y_i - E[Y])^2 - 2a(y_i - E[Y])(x_i - E[X]) + a^2 (x_i - E[X])^2 \\
&= -2 \sum_i (y_i - E[Y])(x_i - E[X]) + 2a \sum_i (x_i - E[X])^2 = 0 \\
\implies a &= \frac{\sum_i (y_i - E[Y])(x_i - E[X])}{\sum_i (x_i - E[X])^2}
\end{aligned}$$

y_i With Variances σ_i

Assume the measurements y_i have different σ_i

Weigh them according to their variance

Use the following abbreviations

$$s = \sum_i \frac{1}{\sigma_i^2}$$

$$s_{xy} = \sum_i \frac{y_i x_i}{\sigma_i^2}$$

$$s_y = \sum_i \frac{y_i}{\sigma_i^2}$$

$$s_{xx} = \sum_i \frac{x_i x_i}{\sigma_i^2}$$

$$s_x = \sum_i \frac{x_i}{\sigma_i^2}$$

$$\Delta = s s_{xx} = s_x^2$$

How do the Errors in a and b Depend on the Errors in y_i?

Using the new abbreviations, we have

$$\sigma_F^2 = \sum_i \sigma_i^2 \left(\frac{\partial F}{\partial y_i}\right)^2$$

$$a = \frac{s_y s_{xx} - s_{xy} s_x}{\Delta}$$

$$b = \frac{s s_{xy} - s_x s_y}{\Delta}$$

$$\sigma_a^2 = \sum_i \sigma_{y_i}^2 \left(\frac{\partial a}{\partial y_i}\right)^2$$

$$\frac{\partial a}{\partial y_i} = \frac{s_{xx} - s_x x_i}{\sigma_{y_i}^2 \Delta} \implies \sigma_a^2 = \frac{s}{\Delta}$$

$$\sigma_b^2 = \sum_i \sigma_{y_i}^2 \left(\frac{\partial b}{\partial y_i}\right)^2$$

$$\frac{\partial b}{\partial y_i} = \frac{s_x x_i - s_x x_i}{\sigma_{y_i}^2 \Delta} \implies \sigma_b^2 = \frac{s_{xx}}{\Delta}$$

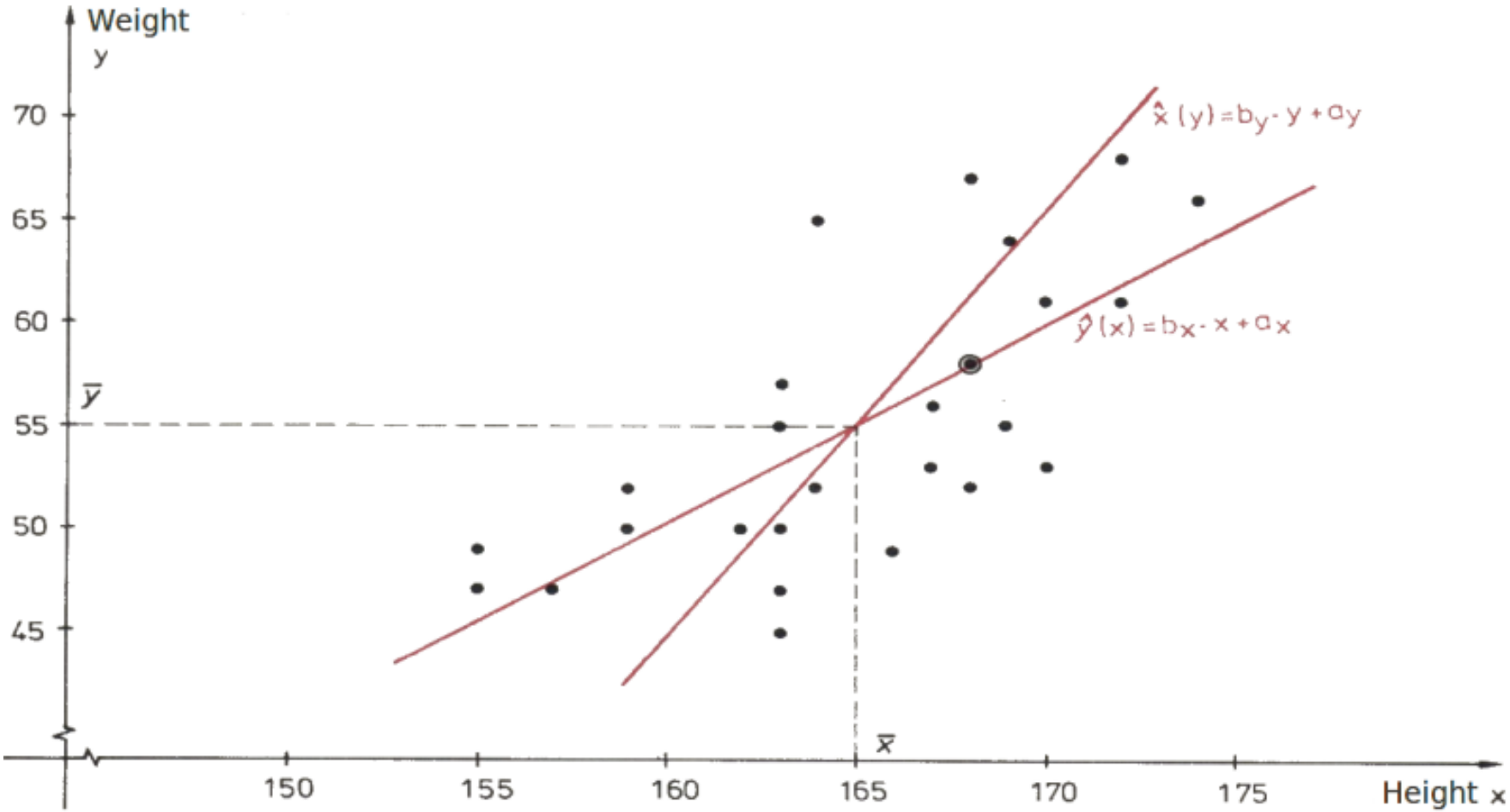
Reverse Argument

By the very same reasoning, we can build $x = by + a$ (the dependency of feature X on feature Y)

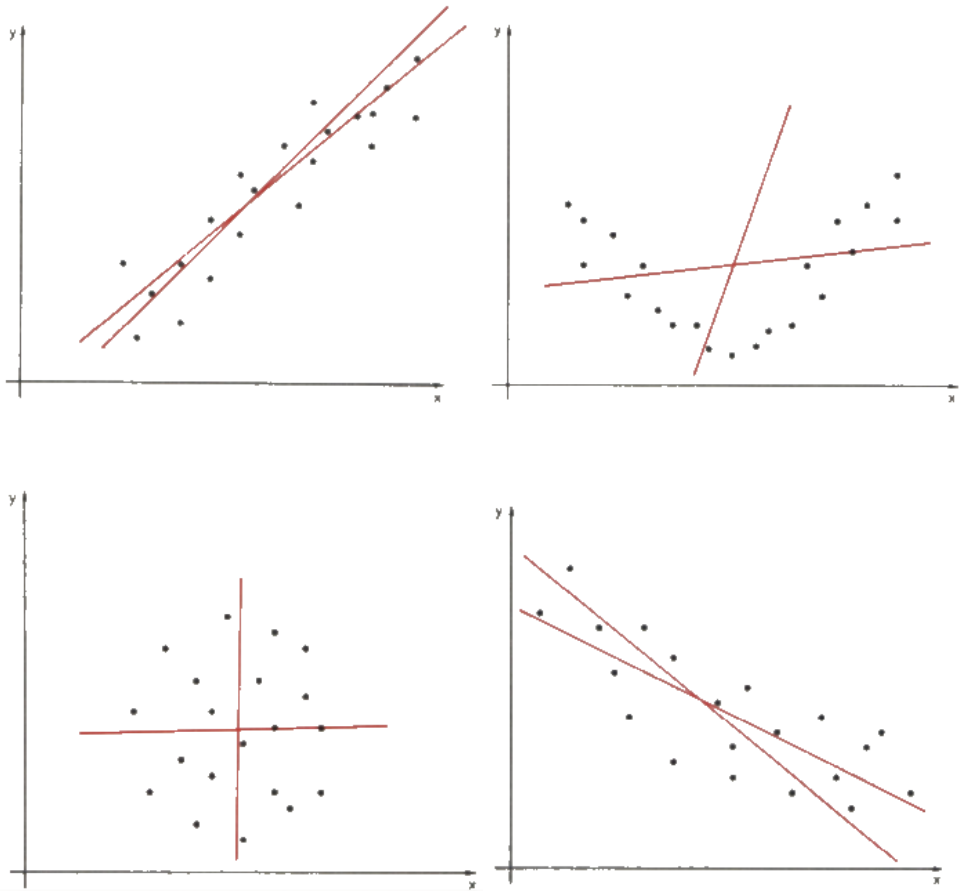
We can subsequently find that

$$a = E[X] - bE[Y]$$
$$b = \frac{\sum_i (y_i - E[Y])(x_i - E[X])}{\sum_i (y_i - E[Y])^2}$$

Both Regression Lines Plotted Together



Possible Intersections



Correlation coefficient

$$r = \frac{\sum_i (x_i - E[X])(y_i - E[Y])}{\sqrt{\sum_i (x_i - E[X])^2}\sqrt{\sum_i (y_i - E[Y])^2}}$$

$$-1 \leq r \leq 1$$

Unit vectors

Scalar product

cos of enclosed angle

Regression Summary

The regression line measures a linear trend in the given data

The covariance is an important part of the regression line

The correlation coefficient indicates how much these trends agree in the given two data series

Various distributions of random variables and their display

Characterizing distributions; central tendency and dispersion

Some theorems on distributions (with/without models)

Standard error of the empirical mean and empirical variance

Error propagation in regression for measurements with precision

→ Hypothesis testing

Hypothesis Testing

Questions

1. How to construct tests?
2. How confident can we be?
3. How many samples do we need (sample size)?

Confidence interval

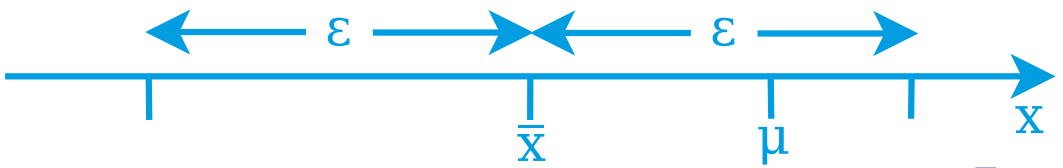
Let X be a random variable, $X_1, ..., X_N$ are samples from X and θ is an unknown parameter. Then, given α, find a **confidence interval** $[\hat{\theta}_L, \hat{\theta}_U]$ such that

$$P(\hat{\theta}_L \leq \theta \leq \hat{\theta}_U) = 1 - \alpha$$

This defines a *confidence level* $1 - \alpha$

Observe

If $x_1, ..., x_n$ are values of $X_1, ..., X_N$, then $\hat{\theta}_L = g_l(x_1, ..., x_n)$ and $\hat{\theta}_U = g_u(x_1, ..., x_n)$ are the bounds of the confidence interval



Required Sample Size to Find μ

Suppose we are given a required width b of a confidence interval at confidence level $(1-\alpha)$ and a distribution X (first normal, then relaxed)

Let X be a random variable, $X_1, ..., X_n$ samples of it, and $\sigma^2 = Var(X) = E[(X-\mu)^2]$. To find $\mu = E[X]$ at confidence level $(1-\alpha)$ and within a confidence interval $\hat{\theta}_U - \hat{\theta}_L \leq 2b$, we distinguish between two cases:

1. Assume $X \sim \mathcal{N}(\mu, \sigma^2)$, σ^2 known

Every $n \geq n_{\min} = \frac{u^2_{1-\alpha/2}\sigma^2}{b^2}$ gives a confidence interval whose width is at most $2b$ and u is a quantile of $\mathcal{N}(0,1)$ at $(1-\alpha/2)$

2. X is arbitrary, σ^2 known

Choose $n \geq n^*_{\min} = \max\{40, n_{\min}\}$. If σ^2 is unknown, replace σ^2 by s^2

Example: Spice Bags

Assume $X_i \sim \mathcal{N}(\mu, 25)$, $\sigma^2 = 25$ known

a) Find a confidence interval at level $(1 - 0.01) = 0.99$ given a sample of size $n = 20$ with $\bar{x} = 99.35g$

$$\hat{\theta}_{L,U} = \bar{X} \pm u_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}$$

Using $n = 20$, $\sigma^2 = 25$, $\bar{x} = 99.35$, and $u_{1-\alpha/2} = u_{0.995} = 2.58$, we have

$$\hat{\theta}_{L,U} = 99.35 \pm 2.58 \frac{5}{\sqrt{20}} = 99.35 \pm 2.88$$

$\Longrightarrow$ the interval is $[96.47, 102.23]$

b) Find $n_{\min}$ using a maximum width of 1 for the confidence interval

$$2b = 1 \implies b = 0.5$$

Using $u_{1-\alpha/2} = 2.58$ and $\sigma^2 = 25$, we have

$$n_{\min} = \frac{u_{1-\alpha/2}^2 \sigma^2}{b^2} = \frac{2.58^2 \cdot 25}{0.25} = 665.64$$

$$\implies n \geq 666$$

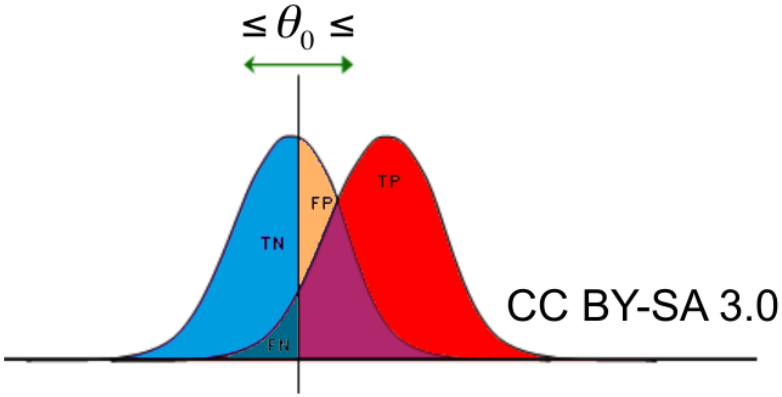
ROC Continuous Example

Classifier: Decide on a disease by varying the diagnosis temperature θ_0

What is the "best" classifier value to indicate that you have the flu?

True positive TP: high temperature and has the flu (positive instance classified as hit)
False negative FN: low temperature, but has the flu (positive instance classified as miss)
True negative TN: low temperature and no flu (negative instance classified as miss)
False positive FP: high temperature, but no flu (negative instance classified as hit)

Classifier prediction (hypothesised class)		Evidence (true class) pos	neg
	hit	TP	FP
	miss	FN	TN



From Confusion Matrix to ROC Plot

Hypothesised class	True class: p	True class: n
y	True positives	False positives (type I error)
n	False negatives (type II error)	True negatives
	$P = TP + FN$	$N = FP + TN$

$$FP \text{ rate } = \frac{FP}{N}$$

$$TP \text{ rate } = \frac{TP}{P} = \text{ recall}$$

$$\text{precision } = \frac{TP}{TP + FP}$$

$$F - \text{score } = \text{precision} \cdot \text{recall}$$

$$\text{accuracy } = \frac{TP + TN}{P + N}$$

A ROC plot has the FP rate on the x-axis and the TP rate on the y-axis

Confusion Matrix

	Condition positive (CP)	Condition negative (CN)	
Test outcome positive (TOP)	True positives	False positives (type I error)	Precision $= \frac{\sum TP}{TOP}$
Test outcome negative (TON)	False negatives (type II error)	True negatives	Negative predictive value $= \frac{\sum TN}{TON}$
	Sensitivity $= \frac{\sum TP}{CP}$	Specificity $= \frac{\sum TN}{CN}$	Accuracy

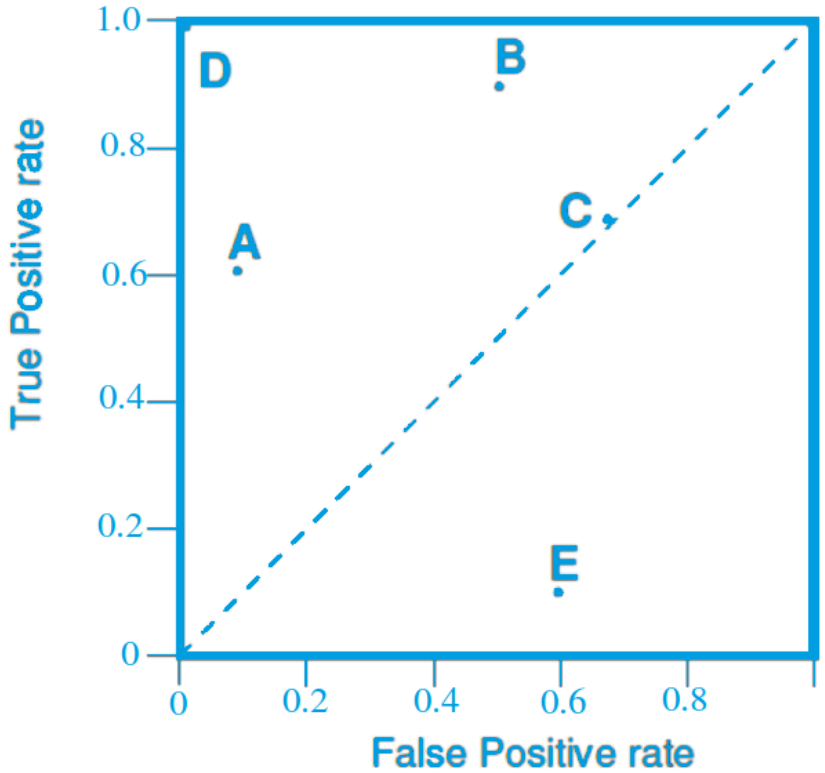
Co Wikipedia (en): "ROC", CC BY-SA 3.0

ROC Plot

ROC plots are two-dimensional graphs in which the TP rate is plotted on the y-axis and the FP rate is plotted on the x-axis

A ROC plot depicts relative trade-offs between benefits (true positives) and costs (false positives)

A discrete classifier is one that outputs only a class label. Each discrete classifier produces a pair (FP rate, TP rate), which corresponds to a single point in the ROC space



A basic ROC plot showing five discrete classifiers

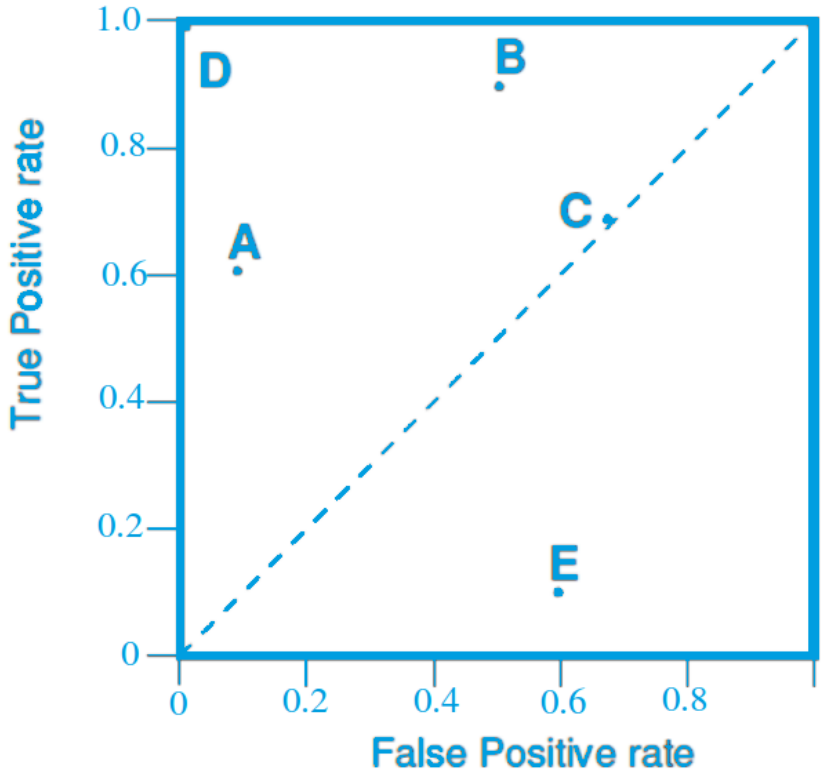
Source: *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*, Tom Fawcett, Intelligent Enterprise Technologies Laboratory, HP Laboratories Palo Alto, slides from GIC09

Important Points in the ROC Space

The lower-left corner $(0, 0)$ represents the strategy of never issuing a positive classification

The opposite strategy is represented by the upper-right corner $(1, 1)$

The point $(0, 1)$ represents perfect classification

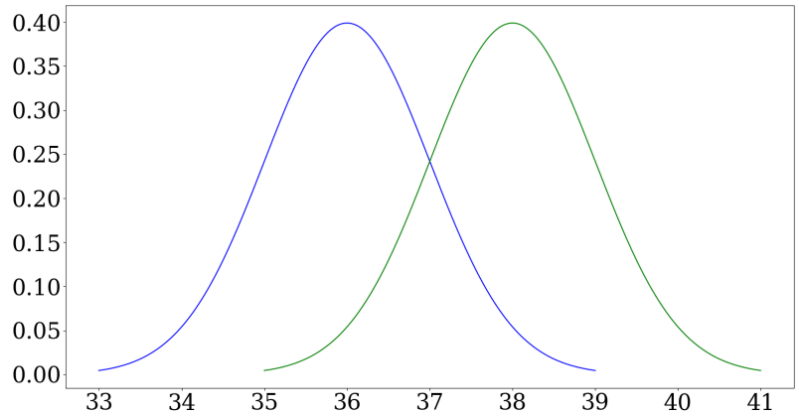


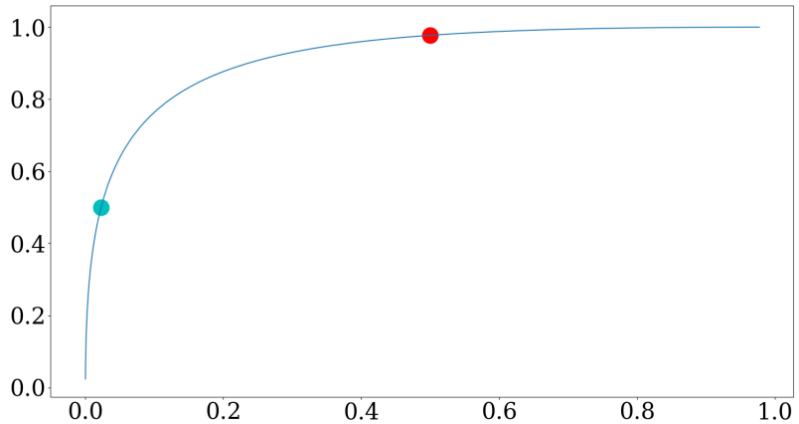
A basic ROC plot showing five discrete classifiers

Source: *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*, Tom Fawcett, Intelligent Enterprise Technologies Laboratory, HP Laboratories Palo Alto, slides from GIC09

ROC Plot: Python Example

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

mu = 36
mu2 = 38
sigma = 1.0
sigma2 = 1.0

x = np.linspace(mu-3, mu+3, 201)
y = np.linspace(mu2-3, mu2+3, 201)

n1 = norm(loc=mu, scale=sigma)
n2 = norm(loc=mu2, scale=sigma2)

plt.plot(x, n1.pdf(x), color='b')
plt.plot(y, n2.pdf(y), color='g')
plt.show()

bounds = np.arange(34, 40, 0.02)
TN = n1.cdf(bounds)
FP = 1. - TN
FN = n2.cdf(bounds)
TP = 1. - FN
TP_rate = TP / (TP + FN)
FP_rate = FP / (FP + TN)
plt.plot(FP_rate, TP_rate)
plt.scatter(FP_rate[100], TP_rate[100], color
      ='r', s=500)
plt.scatter(FP_rate[200], TP_rate[200], color
      ='c', s=500)
plt.show()
```





ROC Plot Interpretation (1/2)

Informally, one point in the ROC space is better than another if it is to the northwest (the TP rate is higher, FP rate is lower, or both) of the first

Classifiers appearing on the left-hand side of an ROC graph, near the x-axis, may be thought of as conservative: they make positive classifications only with strong evidence so they make few false positive errors, but they often have low true positive rates as well

Classifiers on the upper right-hand side of an ROC graph may be thought of as liberal

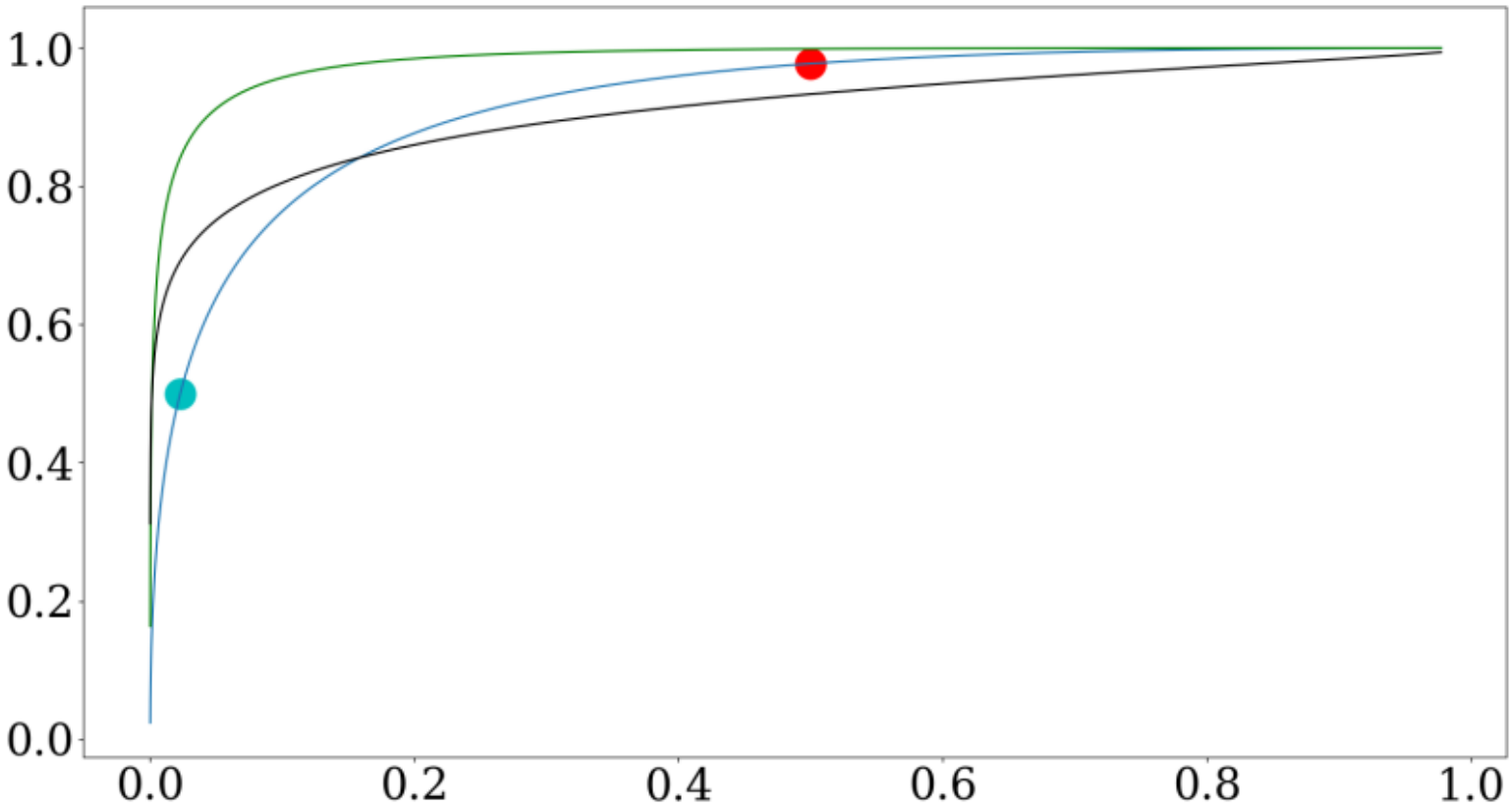
ROC Plot Interpretation (2/2)

The diagonal line $y = x$ represents the strategy of randomly guessing a class

A random classier will produce a ROC point that slides back and forth on the diagonal based on the frequency with which it guesses the positive class. In order to get away from this diagonal into the upper triangular region, the classifier must exploit some information in the data

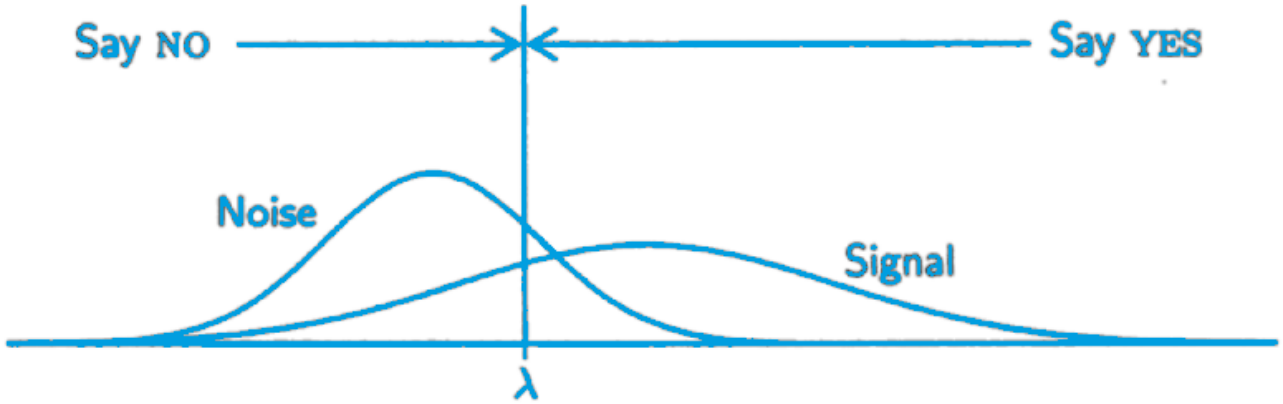
Any classifier that appears in the lower right triangle performs worse than random guessing. This triangle is therefore usually empty in ROC plots

Further ROC Examples



Green curve: ROC curve for $\mu_3 = 39, \sigma_3 = 1.0$
Black curve: ROC curve for $\mu_4 = 39, \sigma_4 = 2.0$

Example: Application to Signals



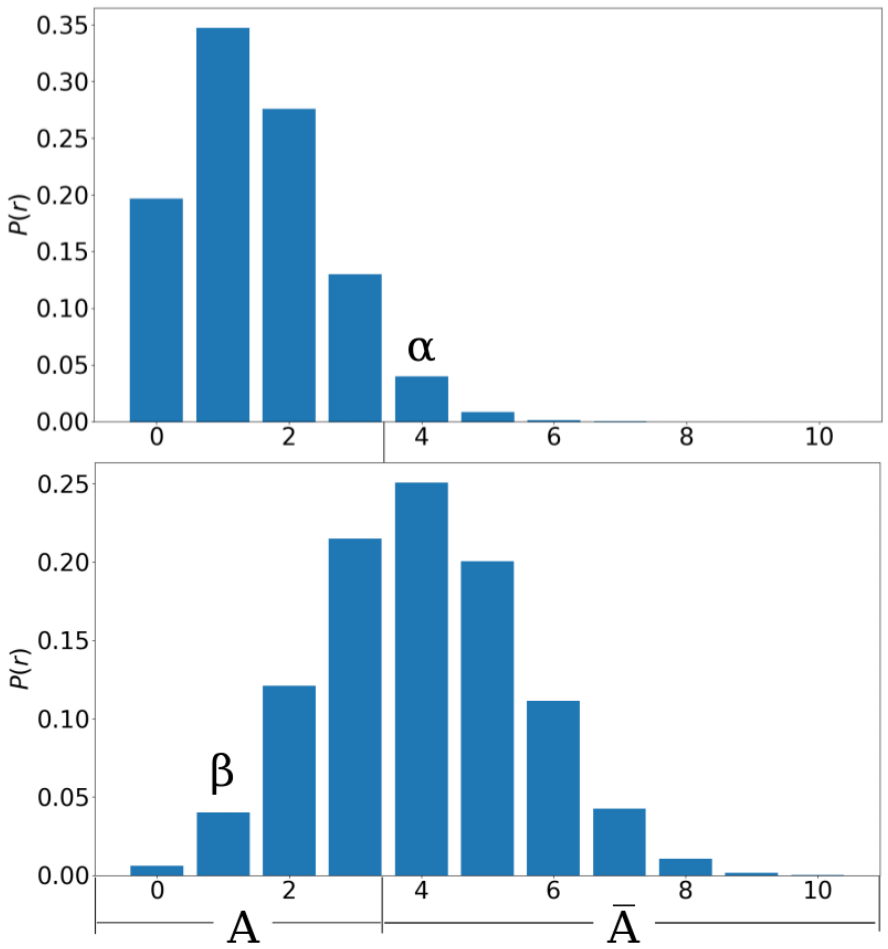
Signal and noise distributions with a decision criterion at λ

X_n: random variable for noise trials, density $f_n(x)$
X_s: random variable for signal trials, density $f_s(x)$

$$false - alarm\ rate : P_F = P(YES|noise)$$
$$= P(X > \lambda) = P(X_n > \lambda) = \int_\lambda^\infty f_n(x)dx$$
$$= 1 - F_n(\lambda)$$

where F_n is the cumulative distribution function of f_n

Test for Two Simple Hypotheses



H_0: screws are defective according to $B(10; 0.15)$
H_1: screws are defective according to $B(10; 0.4)$

Until A: H_0 accepted
Beyond A: H_1 accepted

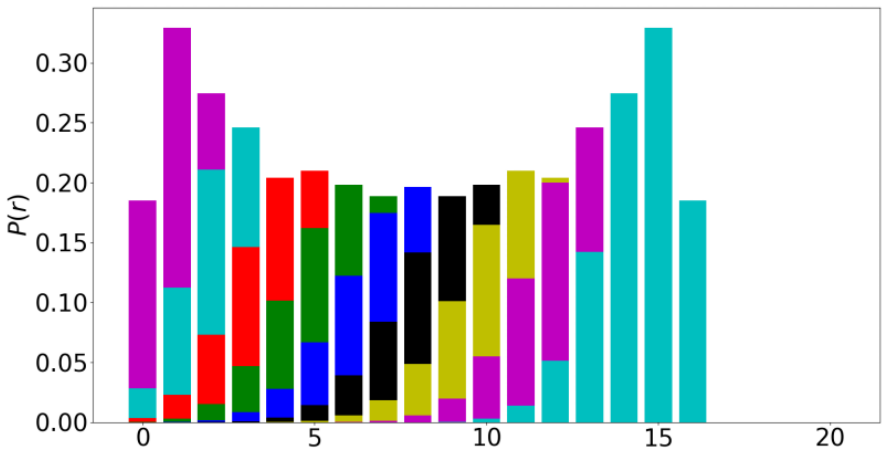
α: type I error
β: type II error

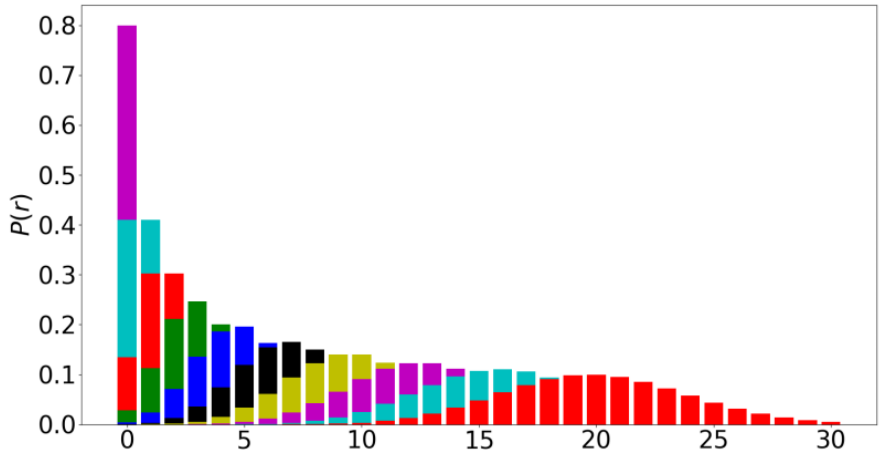
Sample $Z = (0, 1, 1, 1, 0, 0, 1, 1, 1, 0)$

Decision rule

$$\delta_k = \begin{cases} Z \leq k, & \text{decide for } H_0 \\ Z > k, & \text{decide for } H_1 \end{cases}$$

Binomial Examples

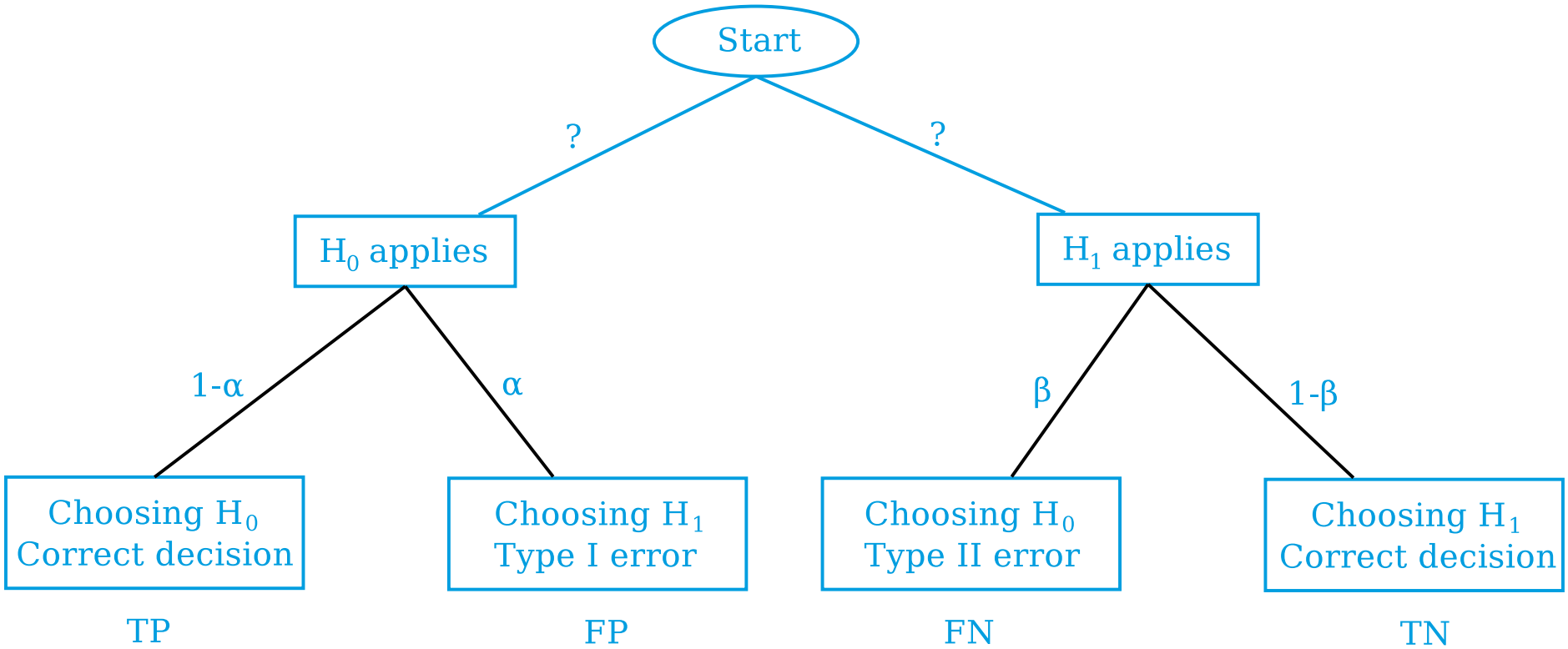





```
import numpy as np
from scipy.stats import binom
import matplotlib.pyplot as plt

colors = ['y', 'm', 'c', 'r', 'g', 'b', 'k',
      'y', 'm', 'c']
x = np.linspace(0, 20, 21)
trials = 16
for i in xrange(0,9):
    p = 0.1 * (i+1)
    plt.bar(x, binom.pmf(x, trials, p), color
          =colors[i])
plt.ylabel('$P(r)$', fontsize=32)
plt.xticks(fontsize=32)
plt.yticks(fontsize=32)
plt.show()
```

```
import numpy as np
from scipy.stats import binom
import matplotlib.pyplot as plt

colors = ['y', 'm', 'c', 'r', 'g', 'b', 'k',
      'y', 'm', 'c']
x = np.linspace(0, 30, 31)
p = 0.2
for i in xrange(0,10):
    plt.bar(x, binom.pmf(x, trials, p), color
          =colors[i])
    trials = (i+1) * (i+1)
plt.ylabel('$P(r)$', fontsize=32)
plt.xticks(fontsize=32)
plt.yticks(fontsize=32)
plt.show()
```

H_0/H_1 Test With Two Simple Hypotheses



Example: Sampling (1/2)

Consider learning the target function "people who plan to purchase new skis this year," given a sample of training data collected by surveying people as they arrive at a ski resort

The instance space X == space of all people (denoted by x). Each individual x may be described by features such as age, occupation, how many times they skied last year, etc.

Some (unknown) distribution D specifies for each person x the probability that x will be encountered as the next person arriving at the ski resort

Example: Sampling (2/2)

The target function $f : X \mapsto \{0, 1\}$ classifies each x according to whether or not they plan to purchase skis this year

We are interested in the following two questions:

1. Given a hypothesis h and a data sample S containing n examples drawn at random according to the distribution D, what is the best estimate of the accuracy of h over future instances drawn from the same distribution?

2. What is the probable error in this accuracy estimate?

Two Definitions of Error

True error

The **true error** of hypothesis h with respect to the target function f and distribution D is the probability that h will misclassify an instance drawn at random according to D

$$error_D(h) = \underset{x \in D}{P}(f(x) \neq h(x))$$

Sample error

The **sample error** of h with respect to the target function f and data sample S is the proportion of examples that h misclassifies

$$error_S(h) = \frac{1}{n} \sum_{x \in S} \delta(f(x) \neq h(x))$$

Here, $\delta(f(x) \neq h(x))$ is 1 if $f(x) \neq h(x)$ and 0 otherwise (decision rule)

How well does $error_S(h)$ estimate $error_D(h)$?

Estimating the Binomial Parameter p is the Same as Estimating $error_D(h)$

Estimate p from a random sample of coin tosses

Toss the coin once

There is a probability p that a coin toss yields heads

$p = \frac{r}{n}$: r heads over n tosses

Estimate $error_D(h)$ from a random sample of instances

Draw a random instance i from D

i is misclassified by h

$error_S(h) = \frac{r}{n}$: r misclassifications over n trials

General Setting for the Binomial Distribution (1/2)

There is a base, or underlying, experiment (e.g., toss of a coin) whose outcome can be described by a random variable, say X. X can take on only two possible values (e.g., $X = 1$ if heads, $X = 0$ if tails)

The probability that $X = 1$ on any single trial of the underlying experiment is given by some constant p, independent of the outcome of any other experiment. The probability that $X = 0$ is therefore $(1 - p)$. Typically, p is not known in advance and the problem is to estimate it

General Setting for the Binomial Distribution (2/2)

A series of n independent trials of the underlying experiment is performed (e.g., n independent coin tosses), producing the sequence of independent, identically distributed random variables $X_1, X_2, ..., X_n$. Let R denote the number of trials for which $X_i = 1$ in this series of n experiments

$$R = \sum_{i=1}^{n} X_i$$

The probability that the random variable R will take on a specific value r (e.g., the probability of observing exactly r heads) is given by the Binomial distribution

$$P(R = r) = \binom{n}{r} p^r (1-p)^{n-r}$$

Problems When Estimating the Error

Bias

If S is a training set, $error_S(h)$ is optimistically biased

$$bias = E[error_S(h)] - error_D(h)$$

For an unbiased estimate, h and S must be chosen independently

Variance

Even with an unbiased S, $error_S(h)$ may still vary from $error_D(h)$

A hypothesis h misclassifies 12 of the 40 examples in S

$$error_S(h) = \frac{12}{40} = 0.30$$

What is $error_D(h)$?

Estimators

Experiment

1. choose a sample S of size n according to a distribution D

2. Measure $error_S(h)$

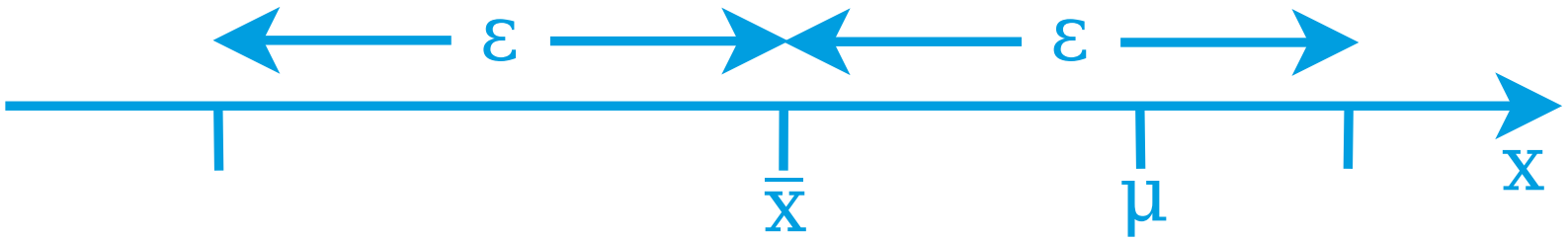
$error_S(h)$ is a random variable and an unbiased estimator of $error_D(h)$

Given the observed $error_S(h)$, what can we conclude about $error_D(h)$?

Confidence Intervals (1/2)

If S contains n examples drawn independently of h and each other and $n \geq 30$, then, with approximately 95% probability, $error_D(h)$ lies in the interval

$$error_S(h) \pm 1.96\sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

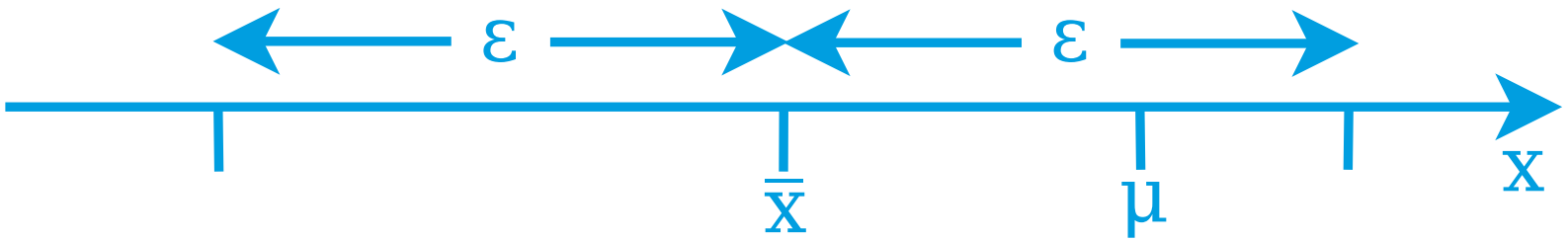


Confidence Intervals (2/2)

If S contains n examples drawn independently of h and each other and $n \geq 30$, then, with approximately $N\%$ probability, $error_D(h)$ lies in the interval

$$error_S(h) \pm z_N \sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

$N\%$	z_N
50%	0.67
68%	1.00
80%	1.28
90%	1.64
95%	1.96
99%	2.58

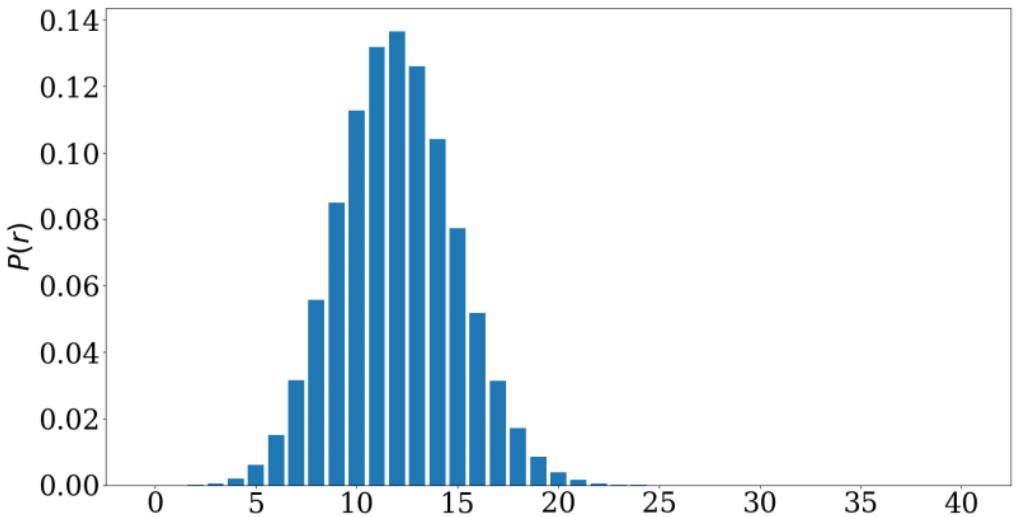


$error_S(h)$ is a (Binomial) Random Variable

Rerun the experiment with different randomly drawn S (of size n)

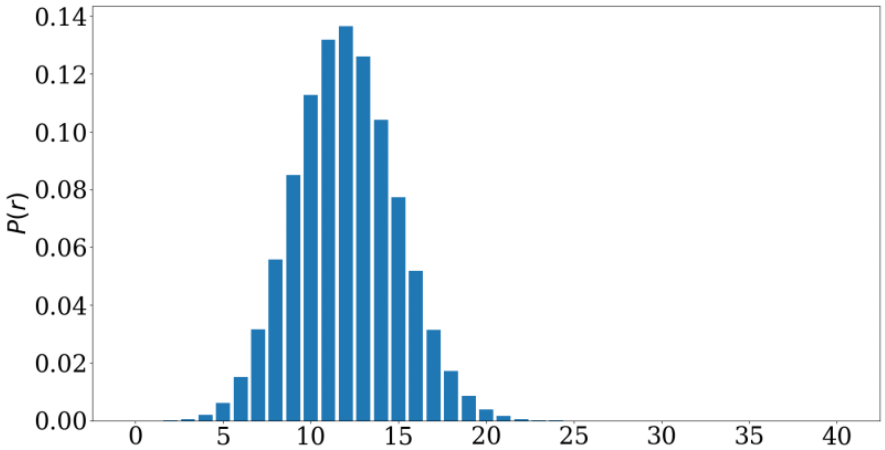
The probability of observing r misclassified examples is given by

$$P(r) = \frac{n!}{r!(n-r)!} error_D(h)^r (1 - error_D(h))^{n-r}$$




```
import numpy as np
from scipy.stats import binom
import matplotlib.pyplot as plt

x = np.linspace(0, 40, 41)
n, p = 40, 0.3
plt.bar(x, binom.pmf(x, n, p))
plt.ylabel('$P(r)$', fontsize=32)
plt.xticks(fontsize=32)
plt.yticks(fontsize=32)
plt.show()
```

Binomial Probability Distribution



If X is binomial, then the probability $P(X = r)$ of r heads in n coin flips is given by $P(r)$

$$P(r) = \frac{n!}{r!(n-r)!} p^r (1-p)^{n-r}$$

Expected value

$$E[X] = \sum_{i=0}^{n} iP(i) = np$$

Variance

$$\begin{aligned} Var(X) &= E[(X - E[X])^2] \\ &= np(1-p) \end{aligned}$$

Standard deviation

$$\begin{aligned} \sigma_X &= \sqrt{E[(X - E[X])^2]} \\ &= \sqrt{np(1-p)} \end{aligned}$$

The Normal Distribution Approximates the Binomial Distribution

$error_S(h)$ follows a **binomial distribution** with mean

$$\mu_{error_S(h)} = error_D(h)$$

and standard deviation

$$\sigma_{error_S(h)} = \sqrt{\frac{error_D(h)(1 - error_D(h))}{n}}$$

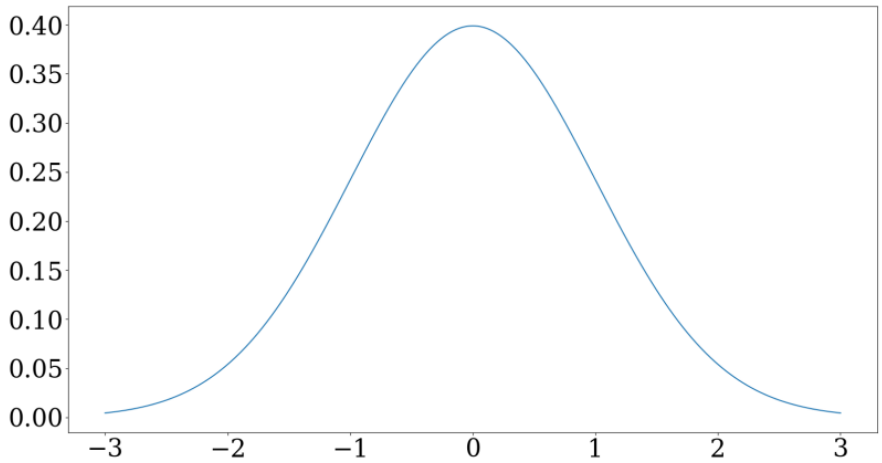
Approximate this by a **normal distribution** with estimated mean

$$\mu_{error_S(h)}$$

and estimated standard deviation

$$\sigma_{error_S(h)} \approx \sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

Normal (Gaussian) Probability Distribution (1/2)



$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

The probability that X will fall into an interval (a, b) is given by

$$\int_a^b p(x)dx$$

Expected value

$$E[X] = \mu$$

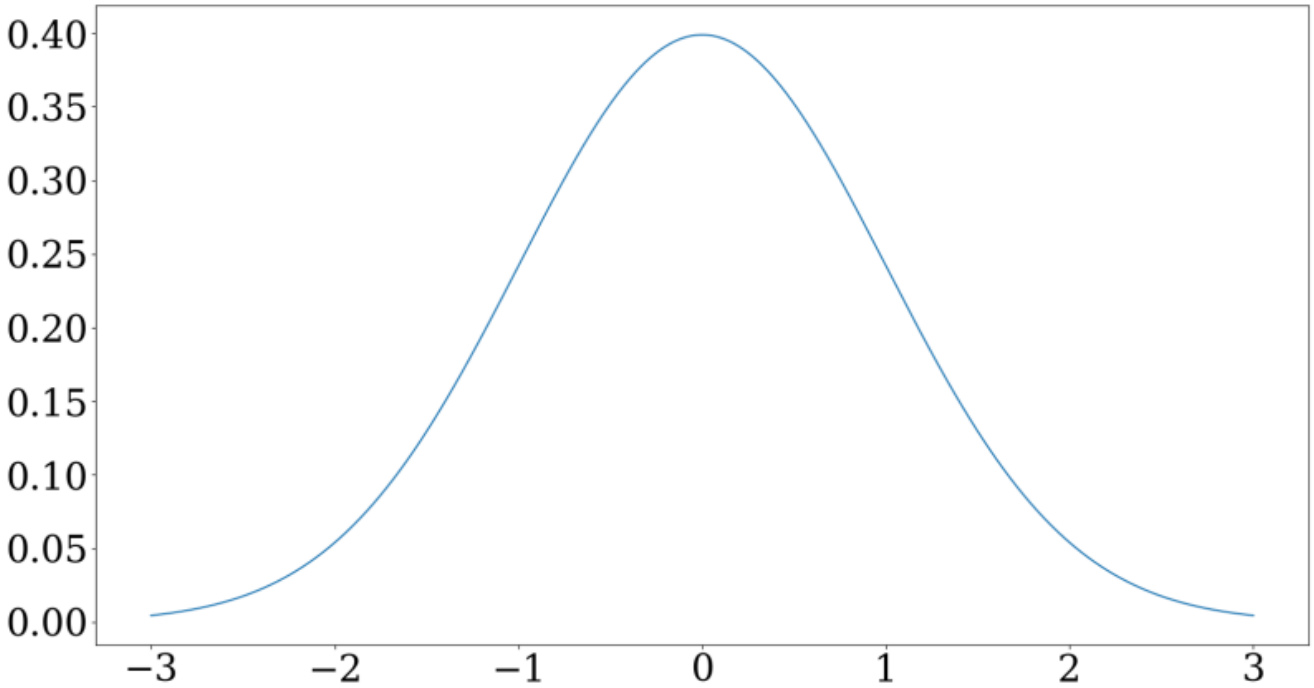
Variance

$$Var(X) = \sigma^2$$

Standard deviation

$$\sigma_X = \sigma$$

Normal (Gaussian) Probability Distribution (2/2)



80% of the area (probability) lies within $\mu \pm 1.28\sigma$

$N\%$ of the area (probability) lies within $\mu \pm z_N \sigma$

$N\%$	50%	68%	80%	90%	95%	98%	99%
z_N	0.67	1.00	1.28	1.64	1.96	2.33	2.58

Confidence Intervals - More Correctly

If S contains n examples drawn independently of h and each other and $n \geq 30$, then, with approximately 95% probability, $error_S(h)$ lies in the interval

$$error_D(h) \pm 1.96\sqrt{\frac{error_D(h)(1 - error_D(h))}{n}}$$

Equivalently, $error_D(h)$ lies in the interval

$$error_S(h) \pm 1.96\sqrt{\frac{error_D(h)(1 - error_D(h))}{n}}$$

which is approximately

$$error_S(h) \pm 1.96\sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

Central Limit Theorem

Consider a set of independent, identically distributed random variables $X_1, ..., X_n$ all governed by an arbitrary probability distribution with mean μ and finite variance σ^2

Define the sample mean

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Central Limit Theorem

As $n \to \infty$, the distribution governing $\bar{X}$ approaches a **normal distribution** with mean μ and variance $\frac{\sigma^2}{n}$

Calculating Confidence Intervals

1. Pick a parameter p that we want to estimate

E.g. $error_D(h)$

2. Choose an estimator

E.g. $error_S(h)$

3. Determine the probability distribution that governs the estimator

$error_S(h)$ is governed by a binomial distribution, which is in turn approximated by a normal distribution (if $n \geq 30$)

4. Find the interval (θ_L, θ_U) such that $N\%$ of the probability mass falls within this interval

Use the table of z_N values

Difference Between Hypotheses

Test h_1 on sample S_1 and h_2 on sample S_2

1. Pick a parameter that we want to estimate

$$d = error_D(h_1) - error_D(h_2)$$

2. Choose an estimator

$$\hat{d} = error_{S_1}(h_1) - error_{S_2}(h_2)$$

3. Determine the probability distribution that governs the estimator

$$\sigma_{\hat{d}} \approx \sqrt{\frac{error_{S_1}(h_1)(1 - error_{S_1}(h_1))}{n_1} + \frac{error_{S_2}(h_2)(1 - error_{S_2}(h_2))}{n_2}}$$

4. Find the interval (θ_L, θ_U) such that $N\%$ of the probability mass falls within this interval

$$\hat{d} \pm z_N \sqrt{\frac{error_{S_1}(h_1)(1 - error_{S_1}(h_1))}{n_1} + \frac{error_{S_2}(h_2)(1 - error_{S_2}(h_2))}{n_2}}$$

Paired t-test for Comparing Two Hypotheses (1/2)

Partition the data into k disjoint test sets $T_1, T_2, ..., T_k$ of equal size, where this size is at least 30

For i from 1 to k, find

$$\delta_i = error_{T_i}(h_1) - error_{T_i}(h_2)$$

Return the value

$$\bar{\delta} = \frac{1}{k} \sum_{i=1}^{k} \delta_i$$

Paired t-test for Comparing Two Hypotheses (2/2)

The $N\%$ confidence interval estimate of d is then

$$\bar{\delta} \pm t_{N,k-1} s_{\bar{\delta}}$$

where

$$s_{\bar{\delta}} = \sqrt{\frac{1}{k(k-1)} \sum_{i=1}^{k} (\delta_i - \bar{\delta})^2}$$

Note that the δ_is are approximately normally distributed

	Confidence level N			
	90%	95%	98%	99%
$\nu = 2$	2.92	4.30	6.96	9.92
$\nu = 5$	2.02	2.57	3.36	4.03
$\nu = 10$	1.81	2.23	2.76	3.17
$\nu = 20$	1.72	2.09	2.53	2.84
$\nu = 30$	1.70	2.04	2.46	2.75
$\nu = 120$	1.66	1.98	2.36	2.62
$\nu = \infty$	1.64	1.96	2.33	2.58

Comparing Learning Algorithms L_A and L_B (1/3)

What we would like to estimate is

$$E_{S \subset D}\left[error_D(L_A(S)) - error_D(L_B(S))\right]$$

where $L(S)$ is the hypothesis output by learner L using training set S, i.e. the expected difference in the true error between the hypotheses output by learners L_A and L_B when trained using randomly selected training sets S drawn according to a distribution D

But what is a good estimator given limited data D_0?

We could partition D_0 into a training set S_0 and a test set T_0 and measure

$$error_{T_0}(L_A(S_0)) - error_{T_0}(L_B(S_0))$$

or even better, we could repeat this process many times and average the results

Comparing Learning Algorithms L_A and L_B (2/3)

1: Partition data D_0 into k disjoint test sets $T_1, ..., T_k$ of equal size (size at least 30)
2: **for** i $\leftarrow 1$ to k **do**
3: use T_i as a test set and the remaining data as a training set S_i
4: $S_i \leftarrow \{D_0 - T_i\}$
5: $h_A \leftarrow L_A(S_i)$
6: $h_B \leftarrow L_B(S_i)$
7: $\delta_i \leftarrow error_{T_i}(h_A) - error_{T_i}(h_B)$
8: **end for**
9: **return** $\bar{\delta} = \frac{1}{k} \sum_{i=1}^{k} \delta_i$

Comparing Learning Algorithms L_A and L_B (3/3)

Notice that we would like to use the paired $t - test$ on δ to obtain a confidence interval

However, the test would not really be correct, as the training sets in this algorithm are not independent (i.e. they overlap!)

It is thus more correct to see this algorithm as producing an estimate of

$$E_{S \subset D_0} [error_D(L_A(S)) - error_D(L_B(S))]$$

instead of

$$E_{S \subset D} [error_D(L_A(S)) - error_D(L_B(S))]$$

But even this approximation is better than making no comparison

Summary (1/3)

Statistical theory provides a basis for estimating the true error $error_D(h)$ of a hypothesis h based on its observed error $error_S(h)$ over a sample S of data. For example, if A is a discrete-valued hypothesis and the data sample S contains $n > 30$ examples drawn independently of h and of one another, then the $N\%$ confidence interval for $error_D(h)$ is approximately

$$error_S(h) \pm z_N \sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

where the values for z_N are taken from a normal distribution table

In general, the problem of estimating confidence intervals is approached by identifying the parameter to be estimated (e.g., $error_D(h)$) and an estimator (e.g., $error_S(h)$) of this quantity. Because the estimator is a random variable (e.g., $error_S(h)$ depends on the random sample S), it can be characterized by the probability distribution that governs its value. Confidence intervals can then be calculated by determining the interval that contains the desired probability mass under this distribution

Summary (2/3)

One possible cause of errors in estimating the accuracy of a hypothesis is *estimation bias*. If X is an estimator of some parameter p, the estimation bias of X is the difference between p and the expected value of X. For example, if S is the training data used to formulate hypothesis h, then $error_S(h)$ gives an optimistically biased estimate of the true error $error_D(h)$

A second cause of estimation error is the variance of the estimate. Even with an unbiased estimator, the observed value of the estimator is likely to vary from one experiment to another. The variance σ^2 of the distribution governing the estimator characterises how widely this estimate is likely to vary from the correct value. The variance decreases as the size of the data sample is increased

Summary (3/3)

Comparing the relative effectiveness of two learning algorithms is an estimation problem that is relatively easy when data and time are unlimited, but more difficult when these resources are limited. One possible approach is to run the learning algorithms on different subsets of the available data, testing the learned hypotheses on the remaining data, then averaging the results of these experiments

In most cases, deriving confidence intervals involves making a number of assumptions and approximations. For example, the confidence interval for $error_D(h)$ involved approximating a Binomial distribution by a normal distribution, approximating the variance of this distribution, and assuming that the instances are generated by a fixed, unchanging probability distribution. While intervals based on such approximations are only approximate confidence intervals, they nevertheless provide useful guidance for designing and interpreting experimental results in machine learning