Today

- Likelihood inference concepts
- Likelihood training algorithm

p-values

- Use constraint; prefer uncertainty intervals
- Some key points (of many)
  - p-value is not the probability that: “null is true”, “data were generated by the null” or “by chance alone”
  - $p < 0.05$ does not mean “the null hypothesis is false”
  - small p-value does not mean “the effect was large or important”
  - $p > 0.05$ does not mean “there was no effect”, or “the null is true”, or “the effect was small”
  - if many replicated studies have $p > 0.05$ it does not provide accumulated evidence of “no effect”

Likelihood in data science

- This week: pure likelihood inference
  - Learning goal: understand likelihood
- Likelihood is also used in
  - Frequentist: as a sample statistic
  - Bayesian: part of the posterior
  - Information theory: e.g. AIC
    - likelihood + complexity penalty

Likelihood

We put all the assumptions about the data generating process in the likelihood function

Likelihood principle

All the evidence in an observation (data) about the parameters (model) is in the likelihood function

---

Likelihood function

Counts all the ways the data could have happened for a given model or hypothesis

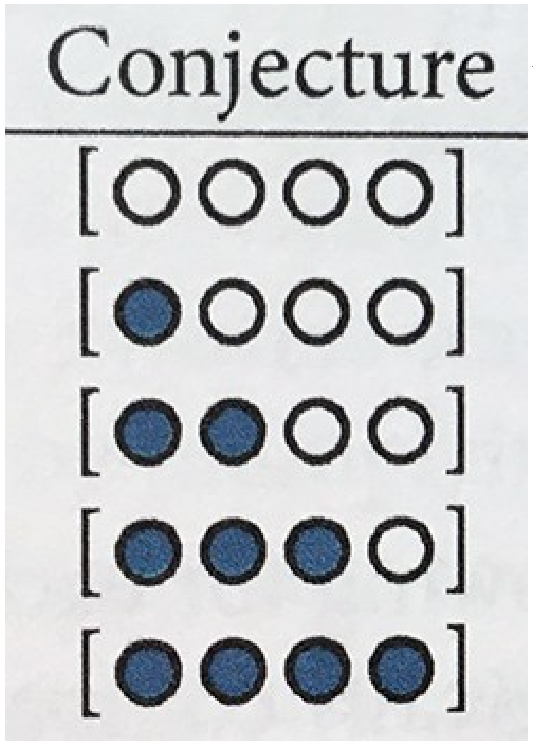
Reading

McElreath Ch 2

Figs 2.2 - 2.4

Marbles in a bag

We know: 4 marbles, 2 colors, marbles drawn randomly with replacement
Goal: what is in the bag?



hypothesis, model

H_1, M_1

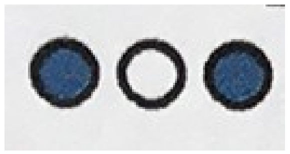
H_2, M_2

H_3, M_3

H_4, M_4

H_5, M_5

Data



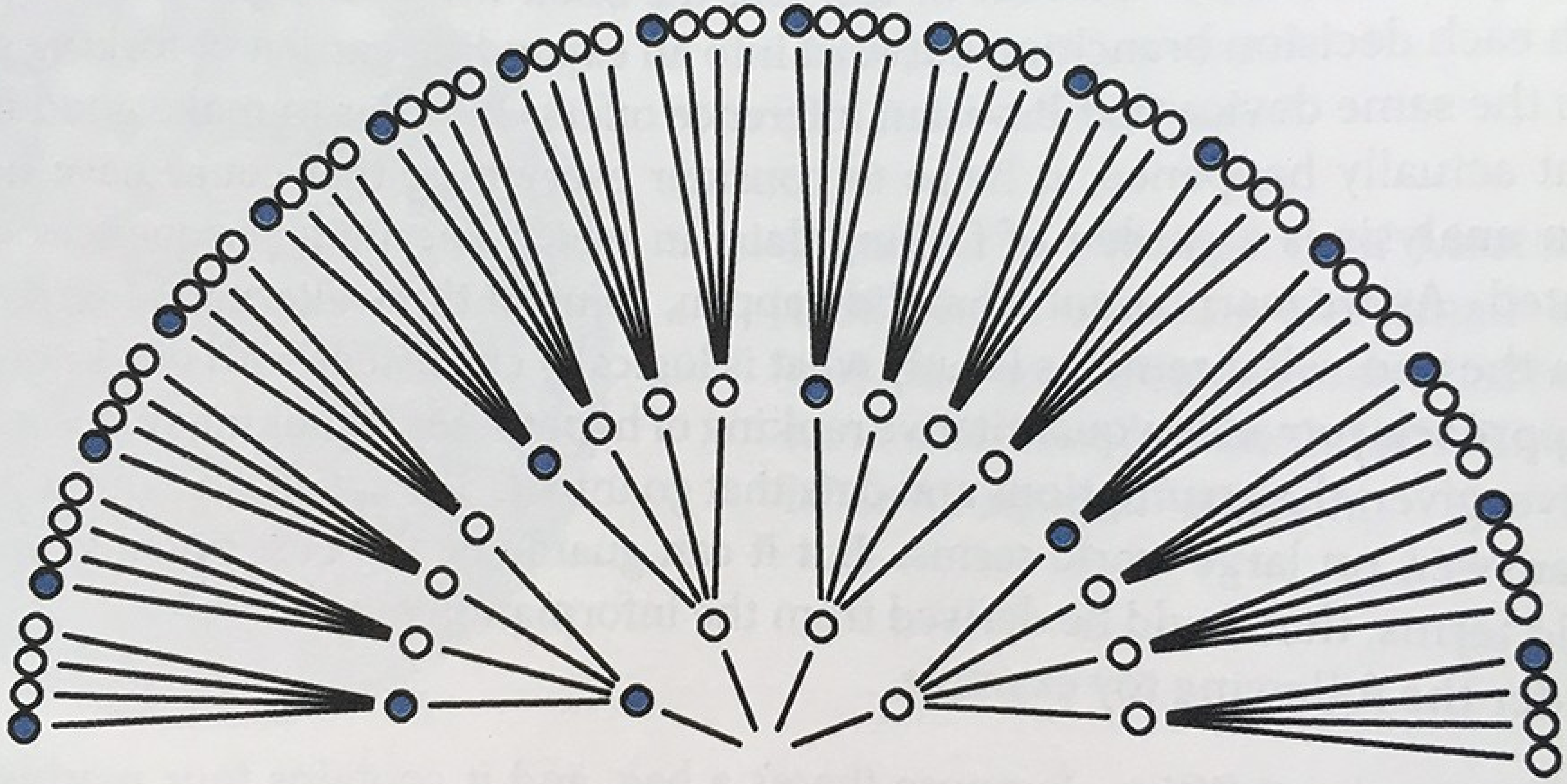


FIGURE 2.2. The 64 possible paths generated by assuming the bag contains one blue and three white marbles.

i.e. assuming we have H_2, M_2 [●○○○]

L = 3/64

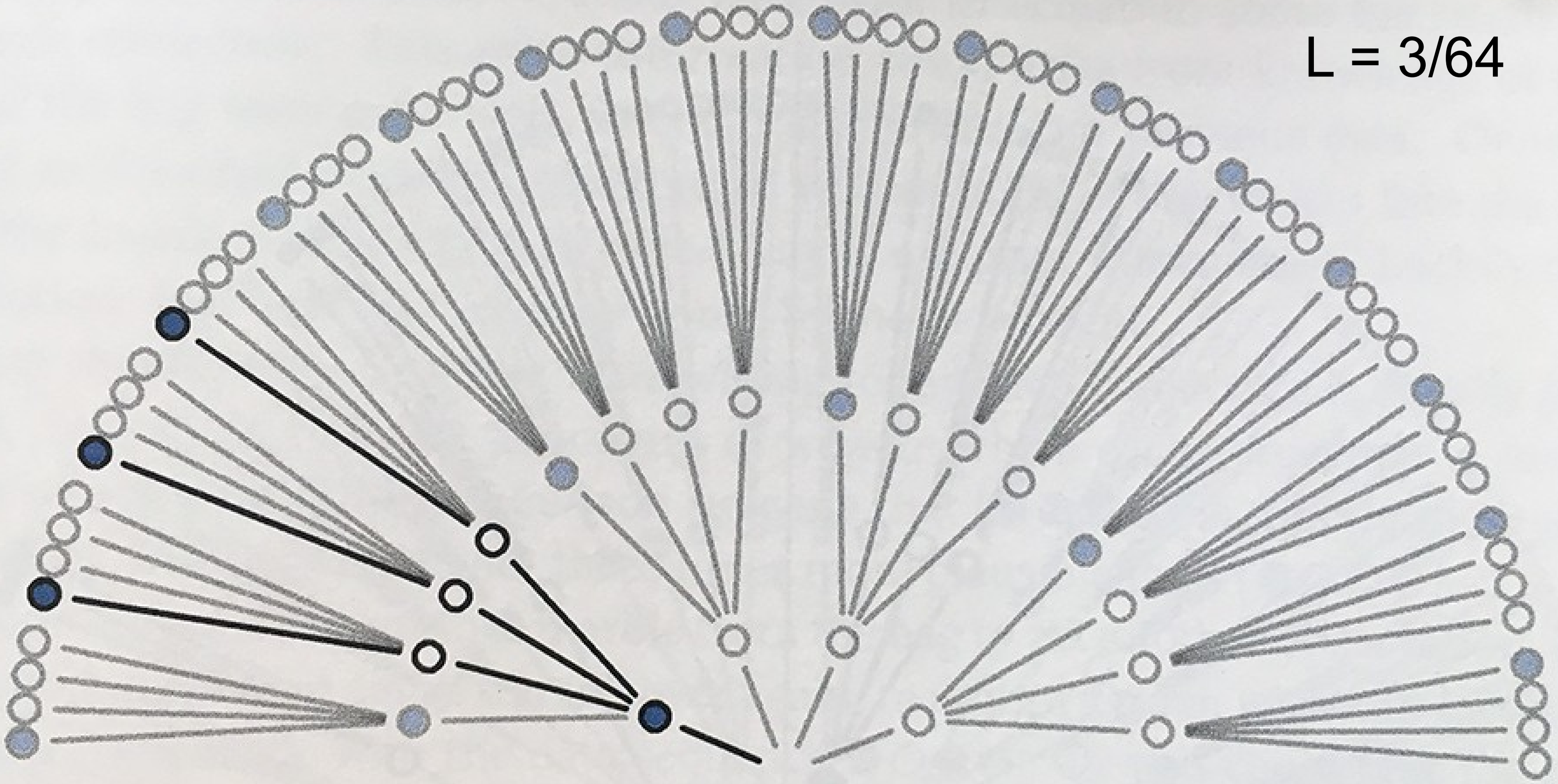


FIGURE 2.3. After eliminating paths inconsistent with the observed sequence, only 3 of the 64 paths remain.

Paths for data ●○● given M_2 [●○○○]

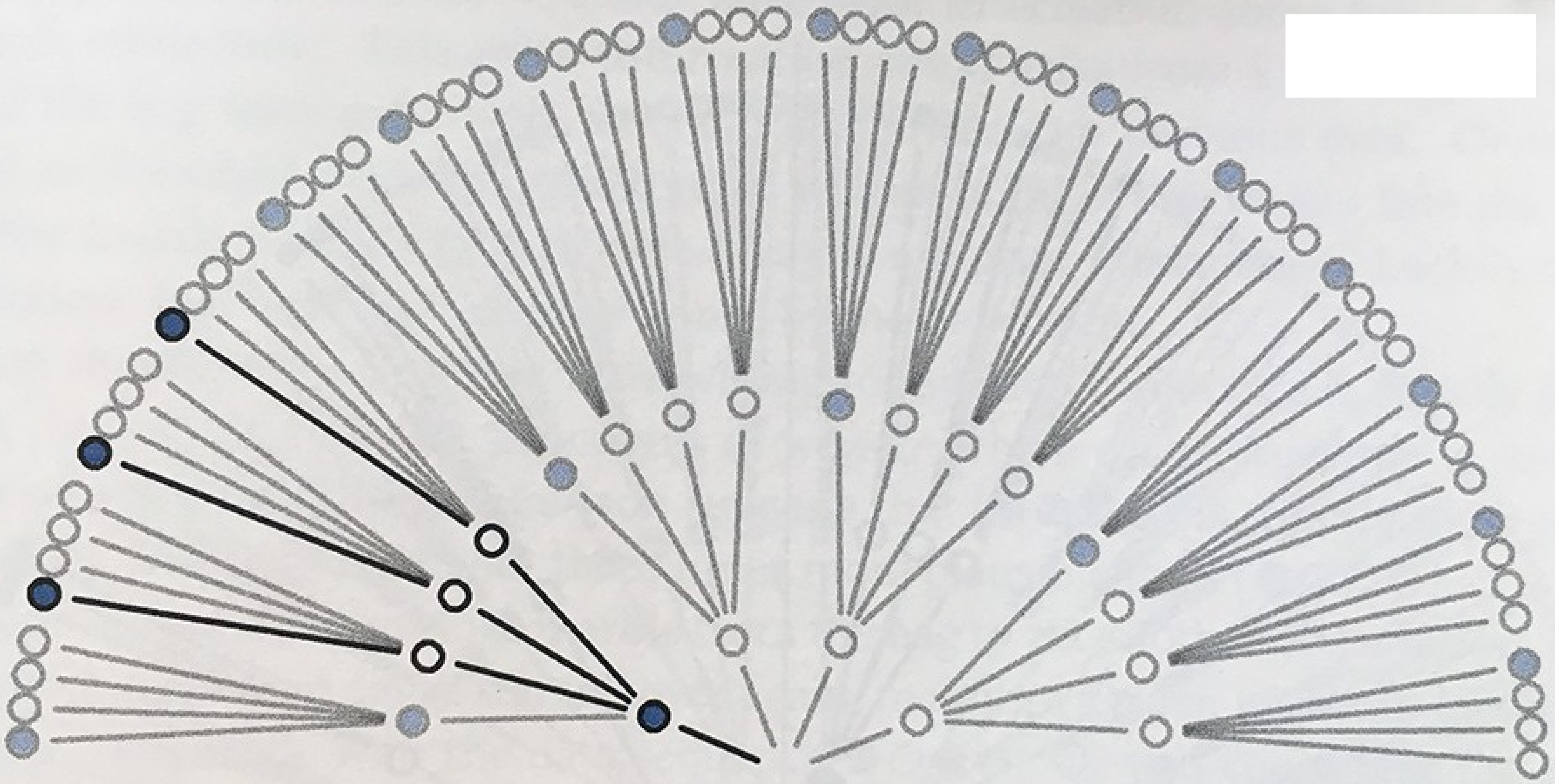
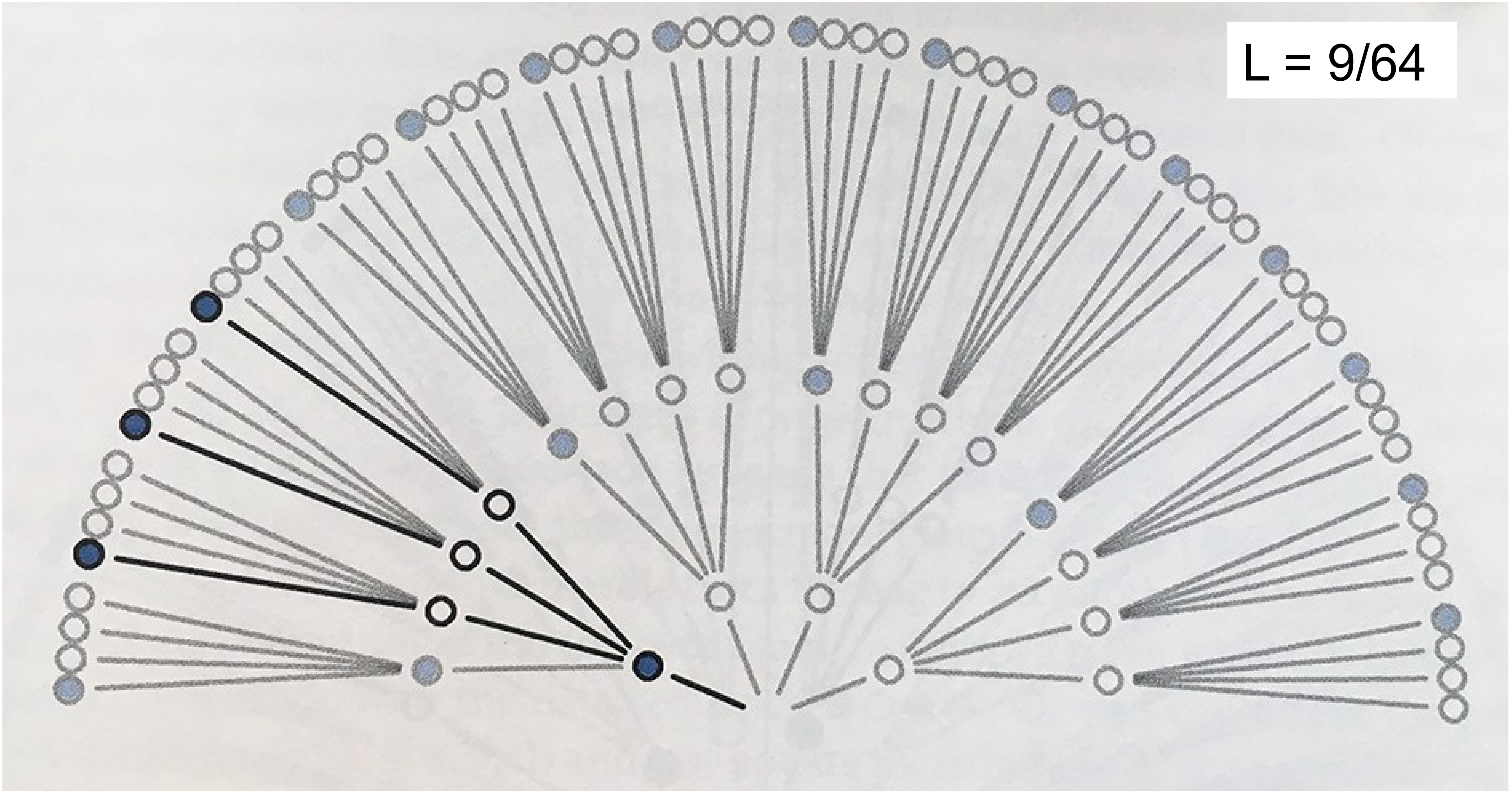


FIGURE 2.3. After eliminating paths inconsistent with the observed sequence, only 3 of the 64 paths remain.

What is the likelihood for
2 blue + 1 white *in any order*?

given M_2 [●○○○]



L = 9/64

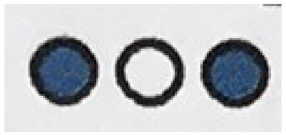
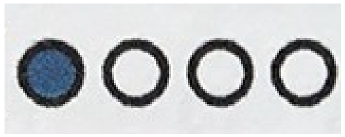
FIGURE 2.3. After eliminating paths inconsistent with the observed sequence, only 3 of the 64 paths remain.

What is the likelihood for
2 blue + 1 white *in any order*?

given M_2 [●○○○]

The likelihood

- Probability of the data given a model

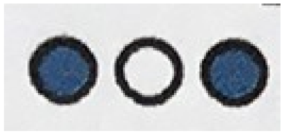
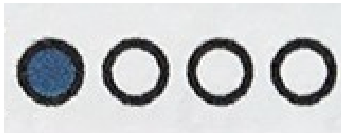
L=P( | )

Data "given" Model 2

A conditional probability

The likelihood

- Probability of the data given a model

L=P( | )

Data ↑ Model 2

"given"

A conditional probability

$$P(y|M_2) = P(y|\theta_2)$$

$y = [\text{"b"},\text{"w"},\text{"b"}]$

could be a vector

θ indicates parameters
(number of blue & white)

Likelihood of model or H

A model is more likely than another if it is the model for which the data are more probable

Notice that this doesn’t mention the *probability* of the model, only the probability of the data

Inference: likelihood ratio

$$\frac{P\left(y|\theta_2\right)}{P\left(y|\theta_1\right)}$$

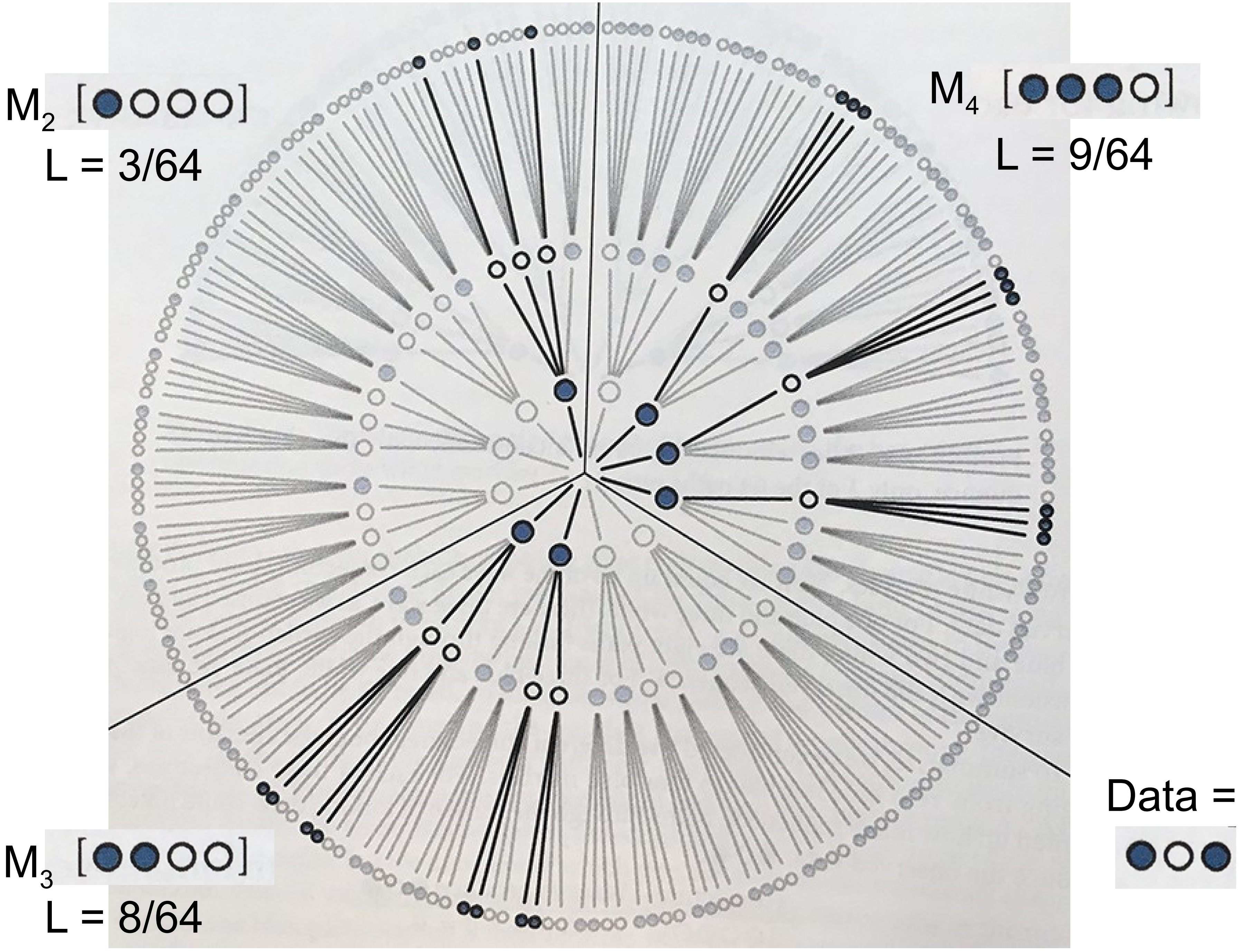
Strength of evidence for model 2 compared to model 1

	Models	Pathways	Likelihood
M_1	[○○○○]	?	
M_2	[●○○○]	3	3/64
M_3	[●●○○]	?	
M_4	[●●●○]	?	
M_5	[●●●●]	?	

Data = ●○●

	Models	Pathways	Likelihood
M_1	[○○○○]	0	0
M_2	[●○○○]	3	3/64
M_3	[●●○○]	?	
M_4	[●●●○]	?	
M_5	[●●●●]	0	0

Data = ●○●


M$_2$ [●○○○]
L = 3/64
M$_4$ [●●●○]
L = 9/64
M$_3$ [●●○○]
L = 8/64
Data =
●○●

	Models	Pathways	Likelihood
M_1	[○○○○]	0	0
M_2	[●○○○]	3	3/64
M_3	[●●○○]	8	8/64
M_4	[●●●○]	9	9/64
M_5	[●●●●]	0	0

Data = ●○●

$$\frac{P(y|\theta_4)}{P(y|\theta_2)}=\frac{9}{3}=3 \qquad \frac{P(y|\theta_4)}{P(y|\theta_3)}=\frac{9}{8}=1.125 \qquad \frac{P(y|\theta_3)}{P(y|\theta_2)}=\frac{8}{3}=2.\dot{6}$$

Notes

- Not frequentist
- Not the same or even similar to a sampling distribution
  - we have not invoked multiple repeated samples
  - probability of the data, not probability of a sample statistic

Likelihood inference for the linear model

Model algorithm

Writing down the model (DGP):

$y_i \sim \mathrm{Normal}(\mu_i, \sigma)$ Stochastic process

$\mu_i = \beta_0 + \beta_1 x_i$ Deterministic process

Model algorithm

Writing down the model:

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \beta_0 + \beta_1 x_i$$

or

$$y \sim \text{Normal}(\mu, \sigma)$$
$$\mu = \beta_0 + \beta_1 x$$

Model algorithm

Writing down the model:

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \beta_0 + \beta_1 x_i$$

or

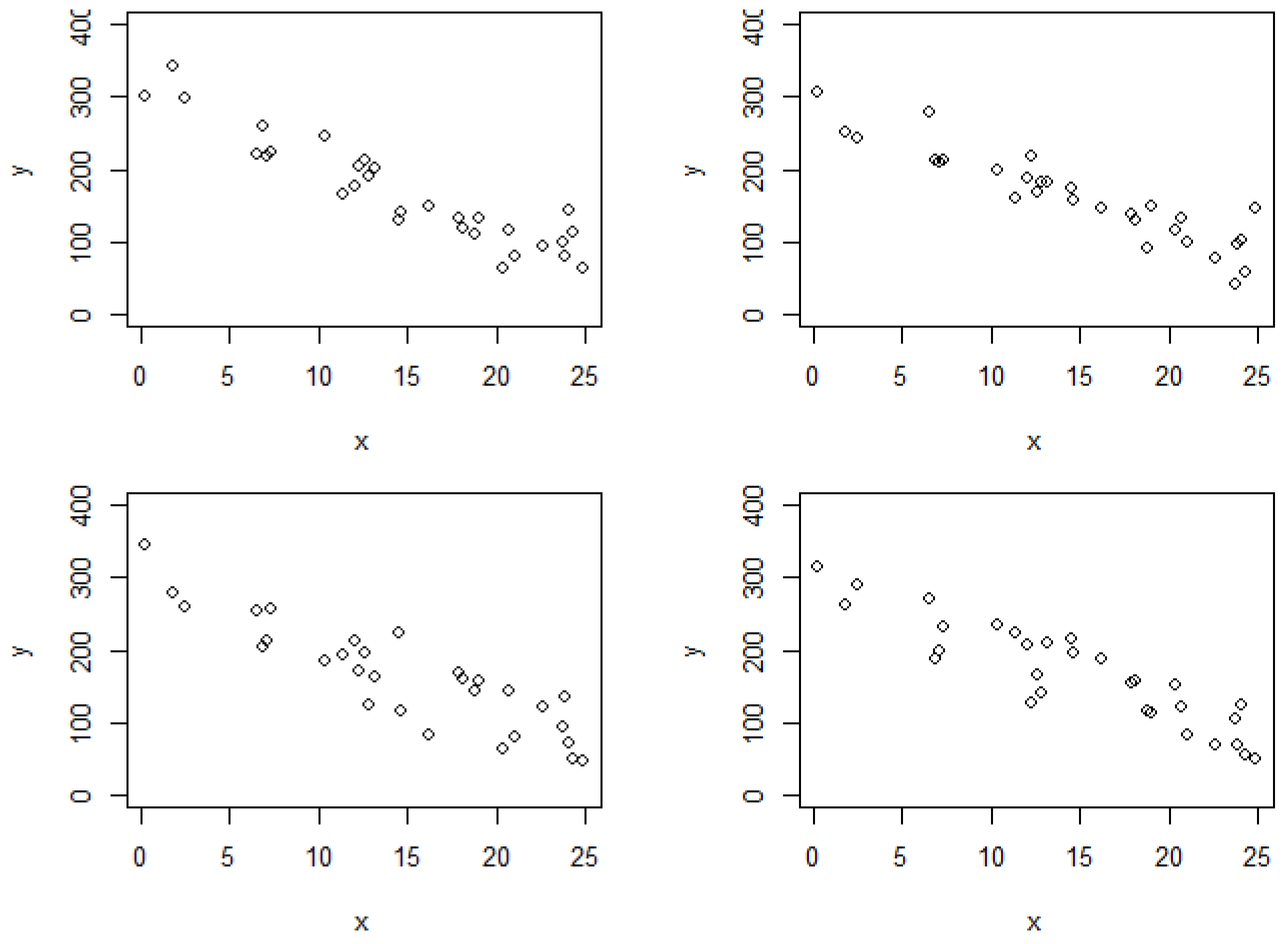
$$y \sim \text{Normal}(\mu, \sigma)$$
$$\mu = \beta_0 + \beta_1 x$$

Algorithm (vectorized):

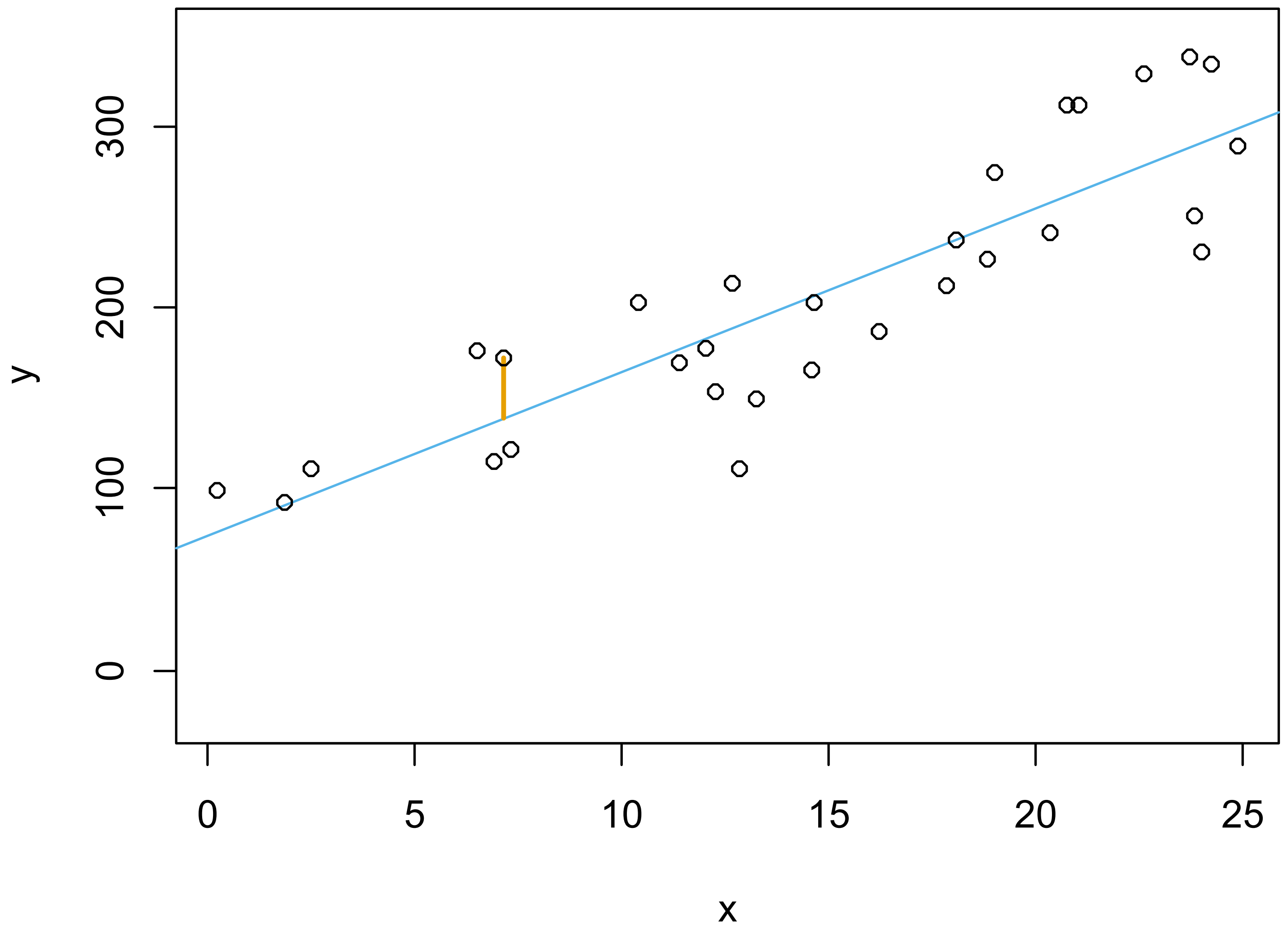
```
lin_skel <- function(beta_0, beta_1, x) {
    return(beta_0 + beta_1 * x)
}
y_stoch <- function(mu=lin_skel(beta_0, beta_1, x), sigma) {
    return(rnorm(n=length(mu), mean=mu, sd=sigma))
}
```

Simulating the model

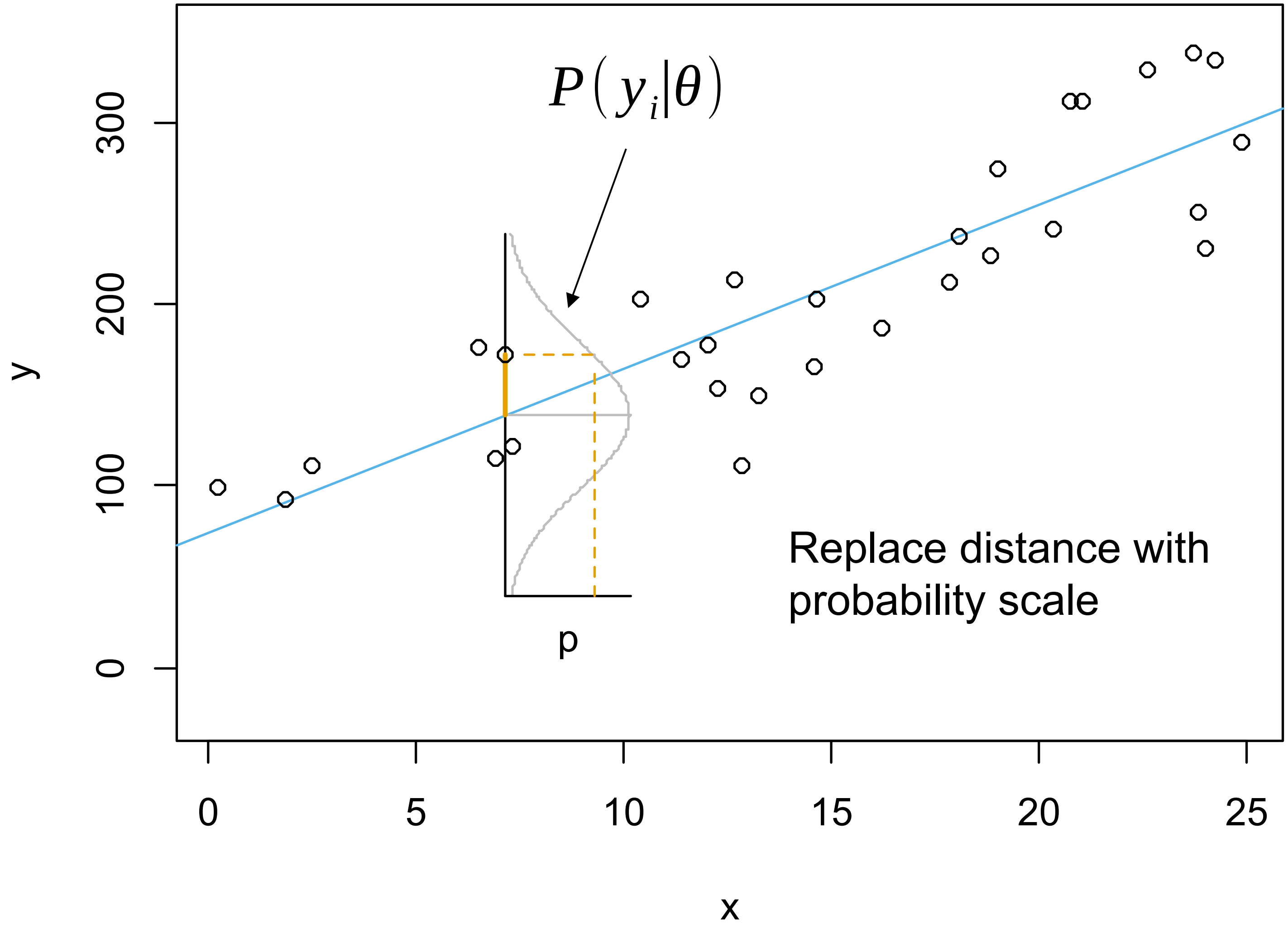
```
y_stoch(mu=lin_skel(beta_0=300, beta_1=-9, x=x), sigma=30)
```



Likelihood (linear, Normal)



Likelihood (linear, Normal)



Likelihood (linear, Normal)

Likelihood for the model:

$$L(\theta) = P(y|\theta) = P(y|\beta_0, \beta_1, \sigma)$$
$$= P(y|\beta_0, \beta_1, \sigma, x)$$

```
dnorm(y, mean=lin_skel(beta_0, beta_1, x), sd)
```

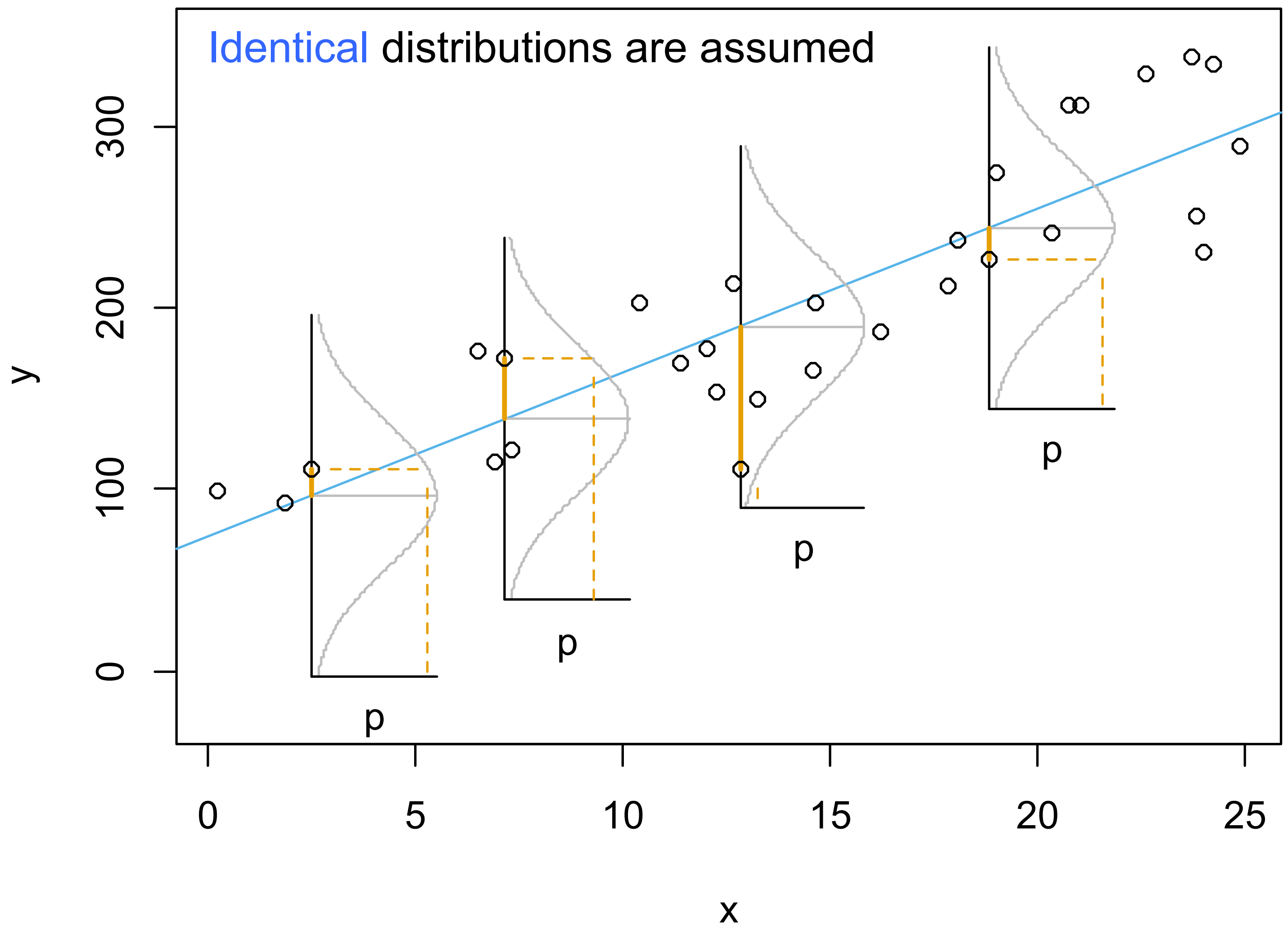
Total likelihood for a data set

One data point: $P\left(y_7 | \theta\right)$

All data points: $\prod_{i}^{n} P\left(y_i | \theta\right)$

because probabilities multiply together to give total probability (*n* is the number of datapoints). Independence is assumed.

Likelihood (linear, Normal)



Support function

The log likelihood:

$$\sum_{i}^{n} \ln P\left(y_i \middle| \theta\right)$$

Instead of multiplying small probabilities, it is more accurate and convenient to sum their logs.

```
sum(dnorm(y, mean=lin_skel(beta_0, beta_1, x), sd, log=TRUE))
```

Training algorithm: Maximum likelihood

The values of the parameters that maximize the likelihood. In other words, the model that maximizes the probability of the data.

An optimization problem.

In practice: minimize the negative log likelihood. The model with the most support, has the smallest negative log likelihood.

Training algorithm

Find maximum likelihood estimates (MLE) for all 3 parameters

```
lin_skel <- function(beta_0, beta_1, x) {
    return(beta_0 + beta_1 * x)
}
```

call the linear model

```
lm_nll <- function(p, y, x) {
    mu <- lin_skel(beta_0=p[1], beta_1=p[2], x=x)
    nll <- -sum(dnorm(y, mean=mu, sd=p[3], log=TRUE))
    return(nll)
}
```

minus sum log likelihoods

```
fit_lm <- optim(p=start_pars, lm_nll, y=y, x=x)
fit_lm
```

Training general approach

- 1) process model function (deterministic skeleton)
  - the biology
  - > `biomod(parameters)`
- 2) nll function (error distribution)
  - the stochasticity or error
  - > `-sum(ddist(data,dpar=biomod,error_parameters,log=T))`
- 3) optimize
  - find biology parameters and error parameters
  - > `optim(parameters, nllfunc, data)`
- This recipe is the same no matter how complicated the process model or what the error distribution is