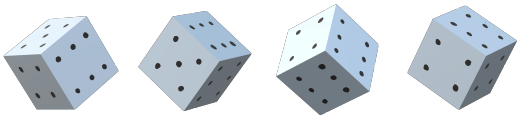
3. Inference

Get experience of formulating questions about a dataset. Be able to assess the accuracy of your inferences, using several different methods.

Inference means reaching conclusions on the basis of data and reasoning. For example, if someone rolls a die and gets 6,4,6,6, what can we conclude about the chance that the next roll is a 6? Do we believe it's ⅙ because that's how dice work? Do we conclude that this die is biased? Do we estimate from the data that the chance of a 6 is ¾, and are we confident enough to make a bet on odds of 3 to 1? How much more confident would we be if we saw the same frequences from a million rolls of the die?



Inference is part science, part philosophy, part craft. The science is computation and probability theory; the philosophy is understanding what questions it is meaningful to ask, and thinking about what you want the answers for; the craft is being able to formulate questions in a way that makes the computing and maths tractable.

3.1. Quantifying a question

The UK Home Office makes available several datasets of police records, at `data.police.uk`. Here is a sample of rows from the stop-and-search dataset.

police force	operation	date-time	lat	lng	gender	age	ethnicity	object of search	outcome
Hampshire	NA	2014-07-31T23:20:00	50.93	-1.38	Male	25–34	Asian	controlled drugs	nothing found
Hampshire	NA	2014-07-31T23:30:00	50.91	-1.43	Male	34+	White	controlled drugs	suspect summonsed
Hampshire	NA	2014-07-31T23:45:00	51.00	-1.49	Male	10–17	White	controlled drugs	nothing found
Hampshire	NA	2014-08-01T00:40:00	59.91	-1.40	Male	34+	White	stolen goods	nothing found
Hampshire	NA	2014-08-01T02:05:00	50.88	-1.32	Male	10–17	White	article for use in theft	nothing found

Suppose we want to investigate possible racial bias in policing. Are the police more likely to stop members of certain ethnic groups? The total number of stops in this dataset is

	Asian	Black	Mixed	Other	White
num. stops	79,492	163,856	350	18,480	483,472

Without knowing context, e.g. population breakdowns in the UK, or typical demographics of people in public spaces, this table is useless. Instead, let's look at the success rates for stop-and-searches. Label each row either find or nothing depending on the outcome of the search. The percentage of stop-and-searches that result in find is

	Asian	Black	Mixed	Other	White
% find	30.0	31.8	60.6	33.1	32.6

The probability of finding criminal activity is lower among Asian suspects, which means that the police are stopping relatively more non-criminals, which is an indicator of racial bias. But is this a significant difference, or is it within the bounds of random variation?

The starting point for quantifying uncertainty is a probabilistic model. Let Y_i be the outcome for row i of the dataset, either find or nothing, and let e_i be the ethnicity of the suspect. The simplest possible model is

$$\mathbb{P}(Y_i = \mathsf{find}) = \beta_{e_i}$$

where β is a vector of probabilities, one per ethnic group. The maximum likelihood estimator is easy to calculate and reassuringly it turns out to be exactly what we would expect from the table: $\beta_{\mathsf{Asian}} = 0.300$, $\beta_{\mathsf{Black}} = 0.318$, etc.

Incorporating features It's extremely unlikely that police behaviour is governed by only one feature in the data. For example, what if the police decision to stop someone is influenced by the suspect's gender as well as ethnicity, and the gender breakdown is different in different ethnic groups?

	Asian	Black	Mixed	Other	White
% Male	96.9	95.2	93.7	93.5	89.4

If a police officer's decision whether or not to stop someone is largely down to the suspect's gender, and if police are relatively more likely to stop male suspects, might this be sufficient to account for the lower $\mathbb{P}(\text{find})$ among Asian suspects? To disentangle the two features, we can propose a model that takes account of both features simultaneously, e.g.

$$\mathbb{P}(Y_i = \text{find}) = \beta_{e_i} + \gamma_{g_i} \tag{9}$$

where g_i is the gender in row i of the dataset. This model allows the probability of find to depend on both ethnicity and gender. If it is indeed gender that is the dominant influence, and if different ethnic groups experience different $\mathbb{P}(\text{find})$ only because of their different gender breakdowns, then the model can accomodate this via $\beta_e = \text{const}$ for all e.

Natural parameters The model (9) has two problems. First, it has too many parameters: we could add 0.1 to every β coefficient and subtract 0.1 from both γ coefficients, and this change would leave absolutely every probability unchanged, and so it is impossible to identify the 'correct' values of the parameters. This issue is known as *non-identifiability*. A common trick is to rewrite the model as

$$\mathbb{P}(Y_i = \text{find}) = \alpha + \beta_{e_i} + \gamma_{g_i}, \quad \text{and require } \beta_{\text{Asian}} = \gamma_{\text{female}} = 0.$$

It doesn't make any difference which reference levels we choose to set to 0; here I chose them alphabetically. We can unwrap this model:

$$\begin{aligned}
&\mathbb{P}(Y = \text{find for Asian female}) = \alpha \\
&\mathbb{P}(Y = \text{find for Asian male}) = \alpha + \gamma_{\text{male}} \\
&\mathbb{P}(Y = \text{find for Black female}) = \alpha + \beta_{\text{Black}} \\
&\mathbb{P}(Y = \text{find for Black male}) = \alpha + \beta_{\text{Black}} + \gamma_{\text{male}} \\
&\dots
\end{aligned}$$

The second problem with (9) is that it allows probabilities that are outside the range $[0, 1]$. We might fix this by changing to a model with explicit truncation,

$$\mathbb{P}(Y_i = \text{find}) = \max\big(0, \min(1, \alpha + \beta_{e_i} + \gamma_{g_i})\big).$$

This truncation turns out to be computationally awkward,[24] when we try to find maximum likelihood parameter estimates. A much better behaved model is

$$\mathbb{P}(Y_i = \text{find}) = \frac{e_i^\xi}{1 + e_i^\xi} \quad \text{where} \quad \xi_i = \alpha + \beta_{e_i} + \gamma_{g_i}. \tag{10}$$

This is just an algebraic gimmick[25] that maps any real number $\xi \in (-\infty, \infty)$ to a value $e^\xi/(1+e^\xi)$ in the range $[0, 1]$. We can just plug the probability formula into a general-purpose

[24] What makes a model computationally awkward? Maximum likelihood estimation is based on optimization. Commonly, optimization libraries work best for functions that are differentiable, and where the partial derivates are only zero at local optima, and where each argument is unconstrained i.e. permitted to take any floating point value. The model with explicit truncation has partial derivates equal to zero over large parts of the parameter space.

[25] We often see medical results like "a Mediterranean diet halves your risk of heart attack". There is usually a model behind this of the form $\mathbb{P}(\text{heart attack}) = e^{\xi+\mu d}/(1+e^{\xi+\mu d})$ where ξ is made up of coefficients relating to other features such as age and gender and weight, μ is a coefficient for the feature "on Mediterranean diet", and $d = 1$ if you follow that diet and 0 otherwise. This sort of study is usually done in populations where the risk of heart attack is fairly small, so the denominator is ≈ 1. For those on the Mediterranean diet $\mathbb{P}(\text{heart attack}) \approx e^\mu e^\xi$ and for those not on it the probability is $\approx e^\xi$. So we can deduce from the headline that the study found the maximum likelihood estimator to be $\hat{\mu} = \log 1/2$. The study won't report what the risk of heart attack was cut *from* or what it was cut *to*, since those numbers depend on ξ which depends on a person's age and gender and weight and so on. The model says "Whatever your underlying risk, your risk would be roughly 50% lower if you were on a Mediterranean diet".

unconstrained optimization routine, it finds the parameters that maximize the likelihood, and whatever parameters it finds we are guaranteed to end up with probabilities in $[0, 1]$. When we unwrap it,

$$\begin{aligned}
&\mathbb{P}(Y = \mathsf{find} \text{ for } \mathsf{Asian\ female}) = e^{\alpha} \,/\, (1 + e^{\alpha}) \\
&\mathbb{P}(Y = \mathsf{find} \text{ for } \mathsf{Asian\ male}) = e^{\alpha+\gamma_{\mathsf{male}}} \,/\, (1 + e^{\alpha+\gamma_{\mathsf{male}}}) \\
&\mathbb{P}(Y = \mathsf{find} \text{ for } \mathsf{Black\ female}) = e^{\alpha+\beta_{\mathsf{Black}}} \,/\, (1 + e^{\alpha+\beta_{\mathsf{Black}}}) \\
&\ldots
\end{aligned}$$

If for example $\gamma_{\mathsf{male}} > 0$, then $\mathbb{P}(Y = \mathsf{find})$ will be higher for male suspects than female suspects, across all ethnic groups. If we compute the maximum likelihood estimates and then unwrap them, we obtain

	Asian	Black	Mixed	Other	White
$\mathbb{P}_{\mathsf{female}}(\mathsf{find})$ %	29.7	31.5	46.0	33.2	32.6
$\mathbb{P}_{\mathsf{male}}(\mathsf{find})$ %	30.2	32.0	46.5	33.7	33.1

The model (10) is called a *logistic regression*. Logistic regression models are in widespread use, for example for estimating the probability that a web user will click on a certain ad. It's up to the data scientist to find good features to put into ξ, for example age and browsing history and purchase history and keywords in emails and location and everything else that a tech company might know about you, plus flashiness and screen size and keywords and everything else that distinguishes the ad.

* * *

We have studiously avoided the question of which model is *true*. The dataset almost certainly has so much richness that any simple parametric model we invent is wrong—but a wrong model can still be useful.

We formulated the logistic regression (10) to answer the question "What is the impact of the suspect's ethnicity on police behaviour, taking account of gender?" The β that we estimate from the model lets us compare ethic groups. If police behaviour is mainly determined by gender, then the β coefficients will be nearly all the same. If police behaviour is mainly determined by some other feature F that we haven't included in the model, then the β coefficients will reflect the breakdown of F in each ethnic group. If ethnicity truly is an influence on police behaviour, then the β coefficients will tell us which ethnic groups have higher $\mathbb{P}(\mathsf{find})$.

Parametric models are a way to ask questions about a dataset. They are one of the best tools we have for asking questions about the dataset, far more subtle than simply tabulating outcomes. But if you give them useless features, they will give useless answers. Garbage in, garbage out.

CONSTRUCTS / LATENT VARIABLES

A *latent variable* is a variable whose value is unobserved (latent means 'hidden'). A latent variable can be something that must exist but we just don't have data for, e.g. the true location of a smartphone user with a noisy GPS. It can also be a *construct*, i.e. a concept constructed in the mind of the data scientist. Latent variables are often useful for linking together different pieces of data, and for explaining our findings in intuitive language. Here is an application, a richer way to investigate possible racial bias in stop-and-search.

What's the difference between a parameter and a latent variable? If the number of unknown quantities grows with the size of your dataset, call them latent variables, otherwise call them parameters.

We set out to investigate whether a suspect's ethnicity influences a police officer's decision to conduct a stop-and-search. Our thought process was this:

> *Build a model for the probability of finding criminal activity, among suspects who were stopped. If the probability of finding criminal activity is lower for suspects in one particular ethnic group, this means that police are stopping more non-criminals in that ethnic group, which is an indicator of racial bias.*

This is a weird model, because it is 'causally backwards'. A person either is or isn't engaged in criminal activity, and this is not an outcome of the police officer's decision to conduct a stop-and-search, so it's weird to build a model for $\mathbb{P}(\mathsf{find})$ for suspects who were stopped.

Seeing as we set out to investigate possible police, how might we use constructs to build a model that explicitly describes the police officer's action and includes a term for bias? Let's invent the construct 'shiftiness'. Let every person in a public space have a shiftiness latent variable s, a floating point number. Suppose it affects two things:

- The higher your shiftiness, the more likely you are to be engaged in criminal activity, e.g. $\mathbb{P}(\text{criminal}) = e^s/(1+e^s)$.
- A police officer will stop you if your shiftiness is above a threshold, say if $s > \alpha + \beta_{\mathsf{ethnicity}} + \gamma_{\mathsf{gender}}$.

This is an invention. It's not trying to be a true measure of some objective feature in criminal psychology. It's just trying to summarize in a variable 'the aggregate of all the various factors and propensities that together affect a police officer's decision to stop-and-search a suspect' so that we can reason more naturally about police behaviour.

Microsoft's Xbox Live uses an invented construct for 'skill of a gamer'[26]. There is a simple probability model: the probability that Player 1 wins against Player 2 in a game is a certain function of the difference in their skill levels. Given a dataset of (player1_id, player2_id, winner) with one record for every game, we can write out

$$\mathbb{P}(\text{dataset of all games and winners}) = f\big(\text{skills of every gamer}\big)$$

and make inferences about each gamer's skill. The results are used to make sure that players are matched with other players of comparable ability.

It can be tricky to design a model with constructs, because of identifiability issues. In the Xbox system, we could add a constant to every single gamer's skill, and it would make no difference to the outcome probabilities. In the police example, we could add a constant to every shiftiness score, and add it to α also, and the distribution would be unchanged. In order to get useful answers out of models with constructs, we need to 'anchor' the values. Bayesian reasoning, described in the next section, is a good way to do this.

[26] TrueSkill, described at `https://www.microsoft.com/en-us/research/project/trueskill-ranking-system` and the subject of an engaging and programmer-friendly blog post `http://www.moserware.com/2010/03/computing-your-skill.html`. The original paper: Ralf Herbrich, Tom Minka, and Thore Graepel. "TrueSkill™: A Bayesian Skill Rating System". In: *NIPS*. 2006. URL: `http://papers.nips.cc/paper/3079-trueskilltm-a-bayesian-skill-rating-system.pdf`.

3.2. Estimating parameters

From the policing data in Section 3.1 we used maximum likelihood estimation to estimate the difference in stop-and-search outcomes between different groups. The difference between male and female suspects was around 0.5% (to be precise, the maximum likelihood estimator was $\hat{\gamma}_{\text{male}} = 0.020711$). How confident should we be in this number? Intuitively, when there are many datapoints (673,541 male suspects, 60,600 female suspects in this case) we should be very confident. If we're looking at a narrower question, e.g. changes in gender bias in stop-and-search incidents in Cambridge city center from week to week, then there will be fewer datapoints and we should be less confident.

In this section we'll look at two different approaches for measuring confidence. We'll work with a toy dataset:

> I have a coin, which might be biased. I toss it $n = 10$ times, and get $k = 9$ heads and $n - k = 1$ tails.

Let's model the data as

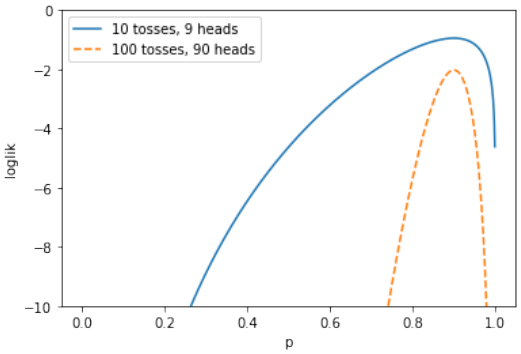
$$\text{num. heads} \sim \text{Binom}(n, p), \quad \text{i.e.} \quad \mathbb{P}(\text{num. heads} = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

which gives the log likelihood function

$$\log \text{lik}(p) = \log \binom{n}{k} + k \log p + (n-k) \log(1-p).$$

Here are plots of the log likelihood as a function of p, for 9 heads out of 10 tosses, and for 90 heads out of 100 tosses. The maximum likelihood estimator is $\hat{p} = 0.9$ in each case. But the plot also shows us the 'tightness' of the maximum, which tells us some sense how confident we should be in rejecting other values of p.

Usually we only care about using the log likelihood to maximize p, so we discard additive constants.



What does it tell us, exactly? There are two main schools of thought, Bayesianism and frequentism.

3.2.1. BAYESIANISM

> *Data science is the process by which we change our beliefs about the world, in the light of data. There's no such thing as objective truth, there's only subjective degree of belief. One should represent belief by using a probability distribution, and one should update it using Bayes' rule.*

Given $k = 9$ heads out of $n = 10$ tosses of a coin, what is the probability of heads?

Prior belief. The probability p of heads is unknown, and Bayesianism requires us to set down a prior belief for it. If we can't quantify our prior belief, Bayesianism says, then there are no grounds for us to draw conclusions. Let's invent, out of thin air, a prior belief that $P \sim \text{Uniform}[0, 1]$, which has density function $f(p) = 1$.

In what follows, we'll write $\Pr(P = p)$ for the density function of P. It's not a probability—since P is a continuous random variable, $\mathbb{P}(P = p) = 0$ for every p. The benefit of this notation rather than $f(p)$ is that Bayesian data science formulae often refer to many different random variables, and it's best to be explicit about which random variable a given density refers to. We'll use the same notation for discrete random variables, where $\Pr(X = x)$ genuinely is $\mathbb{P}(X = x)$. It's convenient to use the same notation for discrete and continuous random variables, because Bayes's rule applies the same to both of them.

Bayesian update. Bayes' rule tells us the density of p, conditional on the observed data. Let K be the random variable 'number of heads'. Then

$$\Pr(P = p \mid K = k) = \frac{\mathbb{P}(K = k \mid P = p)\ \Pr(P = p)}{\int_{q=0}^{1} \mathbb{P}(K = k \mid P = q)\ \Pr(P = q)\, dq}.$$

This is called the *posterior distribution* for P. It's a density function, a function of p, so it must integrate to 1, and so it's convenient to gather up all the terms that don't involve p into a constant and then say "this constant must be whatever it takes to make the posterior density integrate to 1".

$$\begin{aligned}\Pr(P = p \mid K = k) &= \kappa\, \mathbb{P}(K = k \mid P = p)\ \Pr(P = p) \\ &= \kappa \binom{n}{k} p^k (1-p)^{n-k} \\ &= \kappa' p^k (1-p)^{n-k}\end{aligned}$$

where

$$\kappa' = 1 \Big/ \left(\int_{q=0}^{1} q^k (1-q)^{n-k}\, dq \right).$$

This particular density function is the density function of the Beta$(k+1, n-k+1)$ distribution (look it up on Wikipedia). We don't even need to work out κ', all we need to do is recognize the terms involving p.

Draw conclusions. We posed the question "What is the probability of heads?" A Bayesianist says that there's no such thing as the objective bias of the coin, there's only our belief, expressed by the posterior distribution $(P \mid K = k) \sim \text{Beta}(k+1, n-k+1)$. So let's report a 95% confidence interval for $(P \mid K = k)$. A computer can work out the relevant points on the distribution function: run

$$\mathsf{lo,hi} = \mathsf{scipy.stats.beta.ppf([0.025,\ 0.975],\ a}{=}k{+}1,\ \mathsf{b}{=}n{-}k{+}1)$$

and then

$$\mathbb{P}(P \in [\mathsf{lo}, \mathsf{hi}] \mid K = k) = 95\%.$$

We can report all sorts of quantities about the probability of heads. For example, our subjective belief that the coin is biased is

$$\mathbb{P}(P > 0.5 \mid K = k) = 1 - \mathsf{scipy.stats.beta.cdf(0.5,\ a}{=}k+1,\ \mathsf{b}{=}n-k+1).$$

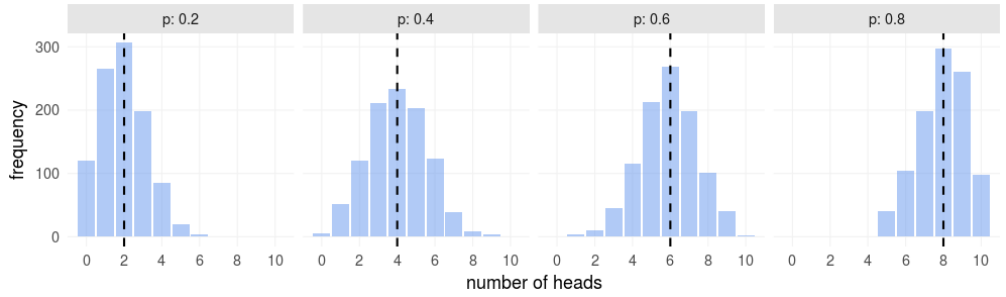
Nuisance parameters. If the problem has many parameters and we only want to report conclusions about one of them, we simply use Bayes's rule to get a joint posterior distribution for all the parameters e.g. $\Pr(P = p, Q = q \mid K = k)$, and then use the law of total probability to find the marginal distribution of the parameter of interest, e.g. $\Pr(P = p \mid K = k) = \int_q \Pr(P = p, Q = q \mid K = k)\, dq$.

3.2.2. FREQUENTISM

> *There is an objective world out there, with fixed but unknown parameters. By observing random phenomena, the data scientist can make inferences about those parameters*

Given $k = 9$ heads out of $n = 10$ tosses of a coin, what is the probability of heads?

Worst-case procedures. The probability of heads, call it p, is fixed and unknown. We can't answer the question directly, and any range we propose for p might be right or wrong, we can't be sure. (Except for the range $[0, 1]$, which is always right and completely useless.) But *whatever* the value of p, simulations of $K \sim \text{Binom}(n, p)$ suggest we're likely to see K in the range $np \pm 2$.



The pivot. An exhaustive computation of $\mathbb{P}(|K-10p| \leq 2)$ over all possible p shows that the lowest value it ever takes is 89%, at $p = 1/2$. We can *pivot* this probability statement:

$$\begin{aligned} & \mathbb{P}\Big(|K-10p| \leq 2\Big) \geq 89\% \qquad \text{for all } p \qquad (11) \\ \Rightarrow \quad & \mathbb{P}\Big(-2 \leq K-10p \leq 2\Big) \geq 89\% \qquad \text{for all } p \\ \Rightarrow \quad & \mathbb{P}\Big(K+2 \geq 10p \geq K-2\Big) \geq 89\% \qquad \text{for all } p \\ \Rightarrow \quad & \mathbb{P}\left(p \in \left[\frac{K-2}{10}, \frac{K+2}{10}\right]\right) \geq 89\% \qquad \text{for all } p. \end{aligned}$$

So, given that we saw 9 heads, can we conclude

$$\mathbb{P}\big(p \in [0.7, 1]\big) \geq 89\% \,?$$

No. The parameter p is fixed and unknown, and it may be inside this range or it may be outside, and we don't know which. What we should really say is that the procedure

```
def confint(k): print(f"p is in [{max(k-2,0)/10}, {min(k+2,10)/10}]")
```

will print a true statement in at least 89% of coin-tossing trials, whatever the value of p, and in this particular trial it happens to print the range $[0.7, 1]$. This is usually abbreviated "An 89% confidence interval for p is $[0.7, 1]$".

Bootstrap resampling. Here is a general-purpose computational method, which removes any need for cleverness or exhaustive optimization for coming up with bounds like (11).

1. Start by writing out the probability you're interested in. Make sure it's a genuine probability, i.e. that there is a random variable inside.
2. Replace any unknown parameters by their maximum likelihood estimates given the data. Replace any random variables by their equivalents drawn from the empirical distribution. This rewritten expression is approximately equal to the probability from step 1.
3. Use the Monte Carlo method to estimate the probability of the expression in step 2.

This is called *bootstrap resampling*. 'Bootstrap' refers to the phrase 'pull yourself up by your bootstraps', in the sense that this method can give us probability answers without our having to even think up a model. 'Resampling' means drawing samples from the empirical distribution, the subject of Section 2.5.

Let's apply it to the problem at hand, 9 heads out of 10 tosses and we want to know the probability of heads. Step 1 says to write out a probability, and it takes some creativity to write down something useful. We'll see more examples in the rest of Section 3 and the example sheet. In this case, we want to make a confidence statement about p, the probability of heads, and a careful look at the pivot from equation (11) suggests that a statement about the maximum likelihood estimator $\hat{p}$ is a useful starting point. Let's try

$$\mathbb{P}\big(\hat{p} \in [p-\delta, p+\delta]\big), \quad \text{where } \delta = 0.1. \qquad (12)$$

Remember that $\hat{p}$ is a function of the data, $\hat{p} = K/n$ in this case if we assume $K \sim \text{Binom}(n, p)$, and so the expression (12) is a genuine probability, as required by Step 1.

The next step is to replace terms. The maximum likelihood estimator (given the data) is $\hat{p} = k/n = 0.9$, so replace p by 0.9 in (12). The $\hat{p}$ term is a random variable, $\hat{p} = K/n$, so replace it by its equivalent drawn from the empirical distribution, K^*/n. There are no

definitive rules about how to do this; in Section 2.5 we saw three different approaches to resampling. The best way to resample is to ask ourselves "If this trial were run again, what is a good way to use the data at hand to synthesize a result that I might plausibly see?" A reasonable approach in this case is to let K^* be the number of heads in 10 values drawn at random from the observed sample, i.e. from 9 heads and 1 tail, which is $K^* \sim \text{Binom}(n, k/n)$. Putting all this together, we have obtained the expression

$$\mathbb{P}\Big(\frac{K^*}{n} \in \big[k/n - \delta, k/n + \delta\big]\Big), \quad \text{where } K^* \sim \text{Binom}(n, k/n). \tag{13}$$

The third step is to use the Monte Carlo method to estimate the probability (13). For $n = 10$, $k = 9$, $\delta = 0.1$, using 10,000 samples, I obtained the answer 92.8%.

```
n,k,δ = 10, 9, 0.1
Kstar = np.random.binomial(n, k/n, size=10000)
np.mean(np.logical_and(Kstar/n >= k/n-δ, Kstar/n <= k/n+δ))
```

Putting all these steps together, and pivoting expression (12) to emphasize p, we get

$$\mathbb{P}\big(p \in [\hat{p} - 0.1, \hat{p} + 0.1]\big) \approx 92.8\%, \quad [\hat{p} - 0.1, \hat{p} + 0.1] = [0.8, 1].$$

I have separated this into two separate statements. The probability statement on the left is about a procedure that we could run on any hypothetical dataset, and it uses $\hat{p}$ to signify a random variable. The equality on the right is based on the actual value of $\hat{p}$ that we get from the actual dataset.

Caveat programmator. Bootstrap resampling is a universal approximation technique. If you invent an unhelpful probability statement in step 1, or if you use a dodgy resampling method for step 2, you might end up with a useless answer. You always need to do a sanity check in your head and ask yourself "For the dataset and question at hand, is there any step in the approach I've taken that will likely give me nonsensical answers?" A data scientist keeps this question at the back of her mind, always. Meanwhile, it's a matter of research in theoretical statistics to find out which probability statements and resampling methods work robustly for which types of question.

3.2.3. PERSPECTIVE

I hope this section leaves you uneasy. On one hand, a Bayesianist won't draw any conclusions at all without a prior—but where do we get prior beliefs from, if not data? On the other hand, a frequentist takes a straightforward question and produces such a contorted answer that you feel you need a hot shower to clean your mind afterwards. Is data science a house built on sand?[27]

The pragmatic answer is that both approaches are different ways to account for uncertainty, and often in data science there are several different sources of uncertainty, and it's useful to be able to mix them.

Example (Bayesian hyperparameters). In the Bayesian approach to the coin question, I pick as my prior belief $P \sim \text{Beta}(\delta, \delta)$. This has the neat feature that the posterior belongs to the same family as the prior, $(P|K = k) \sim \text{Beta}(k+\delta, n-k+\delta)$. If $\delta = 1$ then the prior is uniform. But honestly I have no idea what δ should be. I declare δ to be a *hyperparameter*, which is a fancy way of saying "parameter that I don't have a prior for", and I use non-Bayesian criteria to pick a value for it.

Example (Probability as an API). In the frequentist approach to the coin question, I work out that $[0, .8]$ is a 34% confidence interval, and $[0, .9]$ is a 74% confidence interval. I pass this information on to a Bayesian data scientist, who treats it like a distribution function, and uses

[27] Some great minds have gone down fruitless paths trying to understand inference. For an account of the history: Donald Gillies. *Philosophical theories of probability.* Routledge, 2000. And Ian Hacking. *The Emergence of Probability: A Philosophical Study of Early Ideas About Probability Induction and Statistical Inference.* 2nd ed. CUP, 2006.

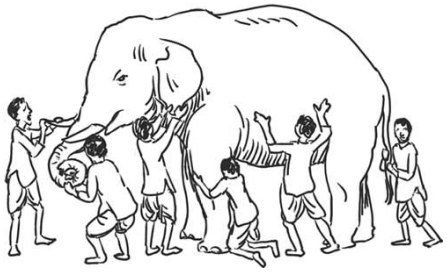
it as a prior distribution for her next analysis. This doesn't make sense, but it gets the job done: I've expressed my uncertainty about the parameter, and she has incorporated uncertainty into her model. We are in effect using the language of probability as a communications API.

Sometimes there is prior data, e.g. someone has conducted a study of "typical bias in coins used in data science textbook illustrations". A Bayesian data scientist might translate those observed frequencies directly into a prior distribution.

Example (Mixed effects modeling). I am analyzing data from a randomized controlled clinical trial, with some subjects taking active medication and some subjects on placebo. In this trial, each subject was assessed on ten visits to the clinic; the condition of patient i on visit j is $X_{i,j}$. I wish to know if there is a systematic difference between the two types of subject.

It's common that the measurements from a single individual are clustered together, so it's not useful to model all the $X_{i,j}$ as independent. Instead, I'll model them using a per-subject construct. Let patient i have a 'wellness score' $\Theta_i \sim \text{Normal}(\mu_{t_i}, \rho^2)$ where $t_i \in \{\mathsf{active}, \mathsf{placebo}\}$, and let $X_{i,j} \sim \text{Normal}(\Theta_i, \sigma^2)$ be independent given Θ_i. This model allows an individual subjects's measurements to be clustered tightly together (if σ is small), and it also allows for a systematic difference between the two types of subject (if $\mu_{\mathsf{active}} \neq \mu_{\mathsf{placebo}}$).

In this model, Θ_i is a parameter for $X_{i,j}$, and we are treating Θ_i as a random variable, which is what Bayesians do. But we can at the same time use maximum likelihood estimation and bootstrap resampling for μ and ρ and σ, like a frequentist. This is called *mixed effects modelling*. The Θ_i are called *random effects* and the other parameters are called *fixed effects*.



The final example is from work by Alan Turing and Irving Good on the Enigma machine[28]. For each message, the German operator would choose a trigraph (sequences of three letters) from a book, the Kenngruppenbuch, which contained all possible trigraphs. The trigraph was used to initialize the wheel positions of the machine, after which the message could be encrypted. Each operator had his own copy of the Kenngruppenbuch, and marked every trigraph that he used and did not re-use it, though it might still be used by other operators. In order to tell the receiver which trigraph was being used, the operator encoded the trigraph using one of nine secret 'digraph tables', with a rule for which table to use on which day; the digraph tables were refreshed once a year or so. The operator would transmit this encoded version of the trigraph, and the receiver would use the digraph table to recover the trigraph. Every day, Bletchley Park had to guess which digraph table was in use that day. Turing devised a method for this, which relied on knowing the distribution of trigraphs. He found, for example, that trigraphs at the top of a page were more likely to be chosen. One step in the calculation was to estimate the probability that a previously unseen trigraph had been chosen. Turing never published his statistical work; it was left to Good to develop the ideas and publish them. Their estimation method is an example of what is now known as *empirical Bayesianism*. Extensions of this method are in use in linguistics (e.g. to estimate Shakespeare's total vocabulary, based on the texts we have of his) and in ecology (to estimate species diversity, based on a sample).

Example (Empirical Bayesianism). I am catching butterflies. Each butterfly species i has frequency θ_i, so the probability that the next butterfly I catch belongs to species i is $\theta_i / \sum_j \theta_j$. What is the probability that the next butterfly I catch is of a species I haven't seen before?

[28] I.J. Good. "Turing's anticipation of empirical Bayes in connection with the cryptanalysis of the naval Enigma". In: *Journal of Statistical Computation and Simulation* (2000). URL: http://dx.doi.org/10.1080/00949650008812016.

Let X_i be the number of butterflies I have seen so far of species i. Let's model $X_i \sim \text{Poisson}(\theta_i)$. The Poisson random variable is a common modeling choice for discrete counts; its mean is $\mathbb{E}X_i = \theta_i$ and its density is $\mathbb{P}(X_i = x) = \theta_i^x e^{-\theta_i}/x!$. If we knew the θ_i, and we knew the total number of species n, then it would be easy to work out the probability of interest:

$$\mathbb{P}\begin{pmatrix}\text{next butterfly}\\ \text{is new species}\end{pmatrix} = \sum_{i=1}^{n} \theta_i 1_{X_i=0} \Big/ \sum_{i=1}^{n} \theta_i. \tag{14}$$

But if we don't know the θ_i and we don't know n, what can we do?

Let's adopt a Bayesian approach and treat the θ_i as random variables drawn independently from some common distribution, say with density function $g(\theta)$, and let Θ be a typical value, $\Pr(\Theta = \theta) = g(\theta)$, and let $X \sim \text{Poisson}(\Theta)$ be a typical count. Then the numerator of (14) is

$$\begin{aligned}\mathbb{E}\Big(\sum_{i=1}^{n} \theta_i 1_{X_i=0}\Big) &= n\,\mathbb{E}\big(\Theta 1_{X=0}\big)\\ &= n\,\mathbb{E}\Big[\mathbb{E}\big(\Theta 1_{X=0} \mid \Theta\big)\Big] \quad \text{by the law of total expectation}\\ &= n\,\mathbb{E}\big(\Theta e^{-\Theta}\big) = n\int_{\theta=0}^{\infty} \theta e^{-\theta} g(\theta)\,d\theta.\end{aligned}$$

This integral involves g and n, which we still don't know. But there is a very clever trick:

$$\mathbb{E}\Big(\sum_{i=1}^{n} 1_{X_i=1}\Big) = n\,\mathbb{E}\big(1_{X=1}\big) = n\mathbb{E}\big[\mathbb{E}(1_{X=1} \mid \Theta)\big] = n\,\mathbb{E}\big(\mathbb{P}(X=1 \mid \Theta)\big) = n\,\mathbb{E}\big(\Theta e^{-\Theta}\big)$$

which suggests we approximate the numerator in (14) by $\sum_i 1_{X_i=1}$, i.e. the number of species for which we have seen exactly one butterfly. Using similar maths, we can approximate the denominator in (14) by the total number of samples we've seen, $\sum_i X_i$. Therefore,

$$\mathbb{P}\begin{pmatrix}\text{next butterfly}\\ \text{is new species}\end{pmatrix} \approx \frac{\text{number of species we've seen once}}{\text{total number of butterflies seen so far}}.$$

What is remarkable in this example is that we used a genuine Bayesian model but without knowing the prior—and we don't actually need to know the prior, because we can extract everything that matters about it from observed frequencies in the data. Large datasets of parallel situations 'describe their own priors'.

* * *

For a grand survey of how data science has been shaped by the interaction of Bayesian and frequentist thinking and by computing resources, see Efron and Hastie[29]. They say

> *A good definition of a statistical argument is one in which many small pieces of evidence, often contradictory, are combined to produce an overall conclusion. In the clinical trial of a new drug, for instance, we don't expect the drug to cure every patient, or the placebo to always fail, but eventually perhaps we will obtain convincing evidence of the new drug's efficacy. The clinical trial is collecting direct statistical evidence, in which each subject's success or failure bears directly upon the question of interest. Direct evidence, interpreted by frequentist methods, was the dominant mode of statistical application in the twentieth century, being strongly connected to the idea of scientific objectivity.*
>
> *Bayesian inference provides a theoretical basis for incorporating indirect evidence [...] The assertion of a prior density $g(\theta)$ amounts to a claim for the relevance of past data to the case at hand.*
>
> *Empirical Bayes removes the Bayes scaffolding. In place of a reassuring prior $g(\theta)$, the statistician must put his or her faith in the relevance of the "other" cases in a large data set to the case of direct interest. [...]*

[29] Bradley Efron and Trevor Hastie. *Computer age statistical inference: algorithms, evidence, and data science*. CUP, 2016. URL: `https://web.stanford.edu/~hastie/CASI/`.

> *The changes in twenty-first-century statistics have largely been demand driven, responding to the massive data sets enabled by modern scientific equipment. Philosophically, as opposed to methodologically, the biggest change has been the increased acceptance of indirect evidence, especially as seen in empirical Bayes and objective ("uninformative") Bayes applications.*

Donald Rumsfeld, the former US Secretary of Defense, famously said[30]

> *Reports that say that something hasn't happened are always interesting to me, because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns—the ones we don't know we don't know.*

Bayesian calculations quantify uncertainty about parameters, and frequentist calculations quantify uncertainty about samples, which are both 'known unknowns'. Wrong models are the 'unknown unknowns'.

[30] U.S. Department of Defense news briefing, 12 February 2002, about the failure to find weapons of mass destruction in Iraq