Problems for Griffiths' Electrodynamics

Contents

481 Lecture 1/16/24

1 Chapter 1: Probablities and Interference (Mackay Ch 2-3)

An ensemble: x random variable

$$A_x = (a_1, a_2, \dots, a_n)$$
$$P_x = (p_1, p_2, \dots, p_n)$$
$$p(x = a_i) = p_i$$

x takes value a_i with probability p_i

$$p \geq 0, \qquad \sum_{a_i \in A_x} p(x = a_i) = 1$$

Short hand for $p(x = a_i)$ is $p(a_i)$, $\quad p(x)$
Joint ensemble: X, Y ensembles

$$XY = \text{ordered pairs}(x, y) \quad x \in A_X, y \in A_Y$$
$$P(x, y) = \text{joint probability of x and y}$$

Marginal probability: $P(x, y) \to P(x), P(y)$

$$P(x) = \sum_{y \in A_y} P(x, y)$$
$$P_x(x = a_i) = \sum_{b \epsilon A_y} P_{XY}(x = a_i, y = b)$$

Conditional probability:

$$P(x = a_i | y = b_j) = \frac{P(x = a_i, y = b_j)}{P(y = b_j)}$$

"Probability of $x = a_i$ given that $y = b_j$ (is true)"

Example 1 $XY = 2$ successive letters in english alphabet. P_x and P_y are identical 'frequency of a letter in english'

$$A_{xy} = \{aa, ab, ac, \dots, zz\}$$

$$P(y | x = 'q')$$

Peak at $y = 'u'$

$$\neq P_Y(y)$$

because x and y are not independent
X, Y "independent" if (and only if) $P(x, y) = P(x)P(y)$
Userful relations: $P(x, y) = P(x|y)P(y) = P(y|x)P(x)$
For any assumption H

$$\forall H : \qquad P(x, y | H) = p(x, y | H)p(y | H)$$

'Sum rule':

$$P(x | H) = \sum_{y \in A_y} P(x, y | H) = \sum_{y \in A_y} P(x | y, H) P(y | H)$$

2 Lecture 1/18

Last time: Main point $P(y|x) \neq P(y)$
Useful relations: Conditional probability

$$P(x|y) = \frac{P(x,y)}{P(y)}$$

where the joint relation is

$$P(x,y) = P(x|y)P(y) = P(y|x)P(x)$$

this can be rewritten into *Baye's theorem*

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

Example 2: Apply Baye's theorem Alex is test for a nast disease.

- Disease status: a (sick or healthy)
- Test outcome: b (positive or negative)

"Test is 95% reliable" or

$$P(+|\text{sick}) = 0.95, \quad P(-|\text{healthy}) = 0.95$$

Disease is nasty but rare $P(\text{sick}) = 0.01$; $P(\text{Healthy}) = 0.99$
Test is positive, what is the probability that Alex is sick? $P(\text{sick}|+) =?$

Solution Use Baye's theorem:

$$P(\text{sick}|+) = \frac{P(+|\text{sick})P(\text{sick})}{P(+)}$$

where $P(+)$ is the probability of a positive test result. This can be found using the sum rule

$$P(+) = P(+|\text{sick})P(\text{sick}) + P(+|\text{healthy})P(\text{healthy})$$

Thus

$$P(\text{sick}|+) = \frac{0.95 * 0.01}{0.95 * 0.01 + 0.05 * 0.99} = 0.161$$

It is useful to write the probabilities in a table

	$b = +$	$b = -$	$P(b)$
$a = \text{sick}$	$0.95 * 0.01$	$0.05 * 0.01$	0.01
$a = \text{healthy}$	$0.05 * 0.99$	$0.95 * 0.99$	0.99
$P(a)$	0.161	0.839	1

where columns represent the 95:5 reliable test.

Exclam!

$$P(S|+) \neq P(+|S)$$

A brief philosphical interlude... The 'Bayesian viewpoint':
Probability as degree of beliefs in propositions given assumptions & evidence, or Probability as 'freq of outcomes in repeat random experiments'

Forward and inverse problems

So far we have talked about Cond Prob, Baye's thrm, and and example.

Generative Model: Parameters $\Theta \rightarrow P(D|\Theta) \rightarrow (P)$ outcomes (data) AKA 'forward problem'
'a model' predicts an outcome given parameters. The model is a probablity distribution due to all the uncertainties and errors we have in the real world.

The Inverse Problem $P(\Theta|D)$
The inverse problem is the opposite of the forward problem (obviously). Also related to the issues regarding 'inference' and using Baye's theorem.

Example 3: A forward problem
An urn contains K balls, B balls are black, and $K - B$ balls are white. A ball is drawn at N times with replacement.

- $n_B = \#$ of times a black ball is drawn
- $P(n_B)$, average n_B?, STD?

With

$$f_B = \frac{B}{K}$$

The probability is given by the binomial distribution

$$P(n_B|N, f_B) = \binom{N}{n_B} f_B^{n_B}(1 - f_B)^{N-n_B}$$

The mean is $N * f_B$ and the STD is $\sqrt{N * f_B * (1 - f_B)}$

Example 4: An inverse problem
We have 11 urns, each with 10 balls. u is the number of black balls in each urn and the urns have $u = 0, 1, \ldots, 10$ black balls. Alex selects an urn at random and draws N balls at random with replacement. Bob wates Alex, but does not know which urn u was selected. For Bob, what is $P(u|N, n_B)$?
We have the data, but we are trying to infer the parameter u

Solution Use Baye's theorem

$$P(u|N, n_B) = \frac{P(n_B|u)P(u)}{P(n_B)}$$

where $P(n_B|u)$ is the 'forward' part from Ex 2, $P(u) = 1/11$, and $P(n_B)$ is the 'normalization' that makes it a valid prob. distribution:

$$P(n_B) = \sum_{u'} P(n_B|u')P(u')$$

Therefore

$$P(u|N, n_B) \propto \binom{N}{n_B}\left(\frac{u}{10}\right)^{n_B}\left(1 - \frac{u}{10}\right)^{N-n_B}$$

e.g. $n_B = 3, N = 10$
insert figure 1.2
The (0,0) point is impossible because we picked 3 black balls, and the urn $u = 0$ has no black balls. The same is true for the (10,10) point. The most likely point is $u = 3\ldots$

Exclam! This is known as 'Posterior Probabilty'

- Θ is the parameter
- D is the data
- $P(\Theta)$ is the prior
- $P(D|\Theta)$ is the likelihood: a function of D prob of data given param (sums to 1 over all options for D). As a function of $\Theta \rightarrow$ likelihood of Θ
- $P(\Theta|D)$ is the posterior
- $P(D)$ is the normalization

! **Probability of *data***

! **Likelihood of *parameters***

Role of Prior:

! You can't do inference without making assumptions

Lecture 1/23/24

Last time:

- Forward $p(data|param)$
- Inverse $p(param|data)$

Using Baye's theorem

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{norm}}$$

Note: You can't do inference w/o working assumptions (prior) priors are subjective. From the inverse problem ex from last week: what is the probability that next ball Alex draws is black?

$$P(B) = \sum P(u)P(B|u)$$

Note: Infereince $\neq$ decision/choice of model. Inference is assigning probabilties to hypothesese.

Problem USB Cable frustrations "It takes 3 tries to plug in a USB cable"
During our first try to plug in the cable, we are collecting data. And if its wrong, we 'believe' that the orientation is wrong, thus we flip it believing that the 2nd try is the correct one. But in fact, this is wrong and the 3rd try is the correct one.

How to collect data?

Lecture 1/25/24

---

3 Chapter 2: Probabilities and Interference (Mackay Ch 2-3)

Example 5: Tossing a coin

- 3 times: H, H, H
- 10 times: H, H, ... H

what is the probability of the next toss being H?

Ex 5.1 Coin with freq of heads f_H is tossed N times and n_H heads. What is the probability of the next toss being H? (Ex 4 but with fixed unknown parameter)
Prior: subjective assumption (e.g. could be uniform) then do inference.

Ex 5.2 N tosses, n_H heads. What is the probability that the coin is biased? (Model Comparison)

Lecture 1/30/24

Last time: Simple inference (within a model) where we solve for $p(data|param)$ and now we move on to model comparison!

Ch 2: Model Comparaison Mackay Ch 3 & 28

A coint that is possibly bent has a frequency of heads f_H. For $N = 100$ tosses, $n_H = 90$ heads which is definitely a bent coin (biased).
For the case $N = 100$, $n_H = 55$, we are not sure if the coin is biased or not. The best fit to data is $f_H = 0.55$ we say that it is probably not bent from our intuition.
For the case $N = 10000$, $n_H = 5500$ we believe that the coin is more likely to be 'bent'

Which model? We know that the fair coin model fits the model less than the bent coin model, but we believe that the fair coin model fits the data better than the bent coin model. From "Occam's Razor" (simplicity): Accept the simplest explanation that fits the data. We would prefer the simpler fair coin model since it is simpler. This is mereley a ad hoc rule of thumb. But Bayesian Calculus naturally implements Occam's Razor.

Comparing hypthesis H_o (fair coin) and H_1 (bent coin) Warning! We should choose the hypothesis set before we see the data, otherwise it is cheating!

Big Picture Two levels of inference

- Level 1: Hypothesis set H_o with parameter f_H: Inferring $P(p_a) = ?$
- Level 1: Hypothesis set H_o no params: no inference
- Level 2: Hypothesis set H_o, H_1: Inferring both $P(H_o)$ and $P(H_1)$

2.1 Coin tosses: 1-param model H_1 (L1 inference)
Outcomes: $X = \{a, b\}$ for heads and tails with probabilities p_a and $p_b = 1 - p_a$
Assumption: The prior on p_a is uniform
F Tosses: data = sequence, s = aaba... with $F_a = \#$ of a's and $F_b = \#$ of b's; $F_a + F_b = F$
The model:

$$P(s|p_a, F, H_1) = p_a^{F_a}(1 - p_a)^{F_b}$$

since the tosses are are specific sequence e.g. aaba... From the definition of H_1

$$p_a \in [0 \ldots 1]$$

is equiprobable and the prior tells use that $p(p_a) = 1$

Questions Given a sequence s of F observations, with $\#\ a = F_a$ and $\#\ b = F_b$,

1. What is my posterior belief about p_a? or $P(p_a) = ?$
2. What is the probability that next draw is a?

As this is a inverse problem, we use Baye's theorem

$$P(p_a|s, F, H_1) = \frac{P(s|p_a, F, H_1)P(p_a)|H_1}{P(s|F, H_1)}$$

the bottom takes the full probabilty of the data no matter the value of p_a and is the normalization

$$= \frac{p_a^{F_a}(1-p_a)^{F_b}(1)}{\int_0^1 p_a^{F_a}(1-p_a)^{F_b} dp_a}$$

where we use the sum rule for the denominator

$$\sum_{p_a} P(s|p_a, F, H_1)P(p_a|H_1)$$

but since it is a continuous variable, we use the integral instead of the sum. The math gives us the gamma function

$$\text{normalization factor} = \frac{F_a!F_b!}{(F_a + F_b + 1)!}$$

Examples $s = aba$ vs $s = bbb$

$$P(p_a|s = aba) \propto p_a^2(1-p_a) \quad \text{vs} \quad P(p_a|s = bbb) \propto (1-p_a)^3$$

The first looks like a parabola and the second looks like a decaying cubic function. In each case, the most probable p_a is $2/3$ and 0 respectively which is shown by the data.

Probability of next toss is a We need to integrate over the prior to get the probability of the next toss being a.

$$P(\text{next} = a) = \int \mathrm{d}p_a \, P(\text{next} = a|p_a)P(p_a|s, F, H_1) = \int \mathrm{d}p_a \, P(p_a|s, F, H_1)p_a = \text{average of } p_a$$

the average of p_a for the first example is $3/5 = 0.6$ and for the second example is $1/5 = 0.2$

Conclusion: We found Probability of s given p_a and H_1 (Data given biased coin model) and the probability of p_a given s, F, H_1 (inference), or forward and inverse probabilities for the biased coind model H_1.

2.2 Zero-parameter model H_o (Fair coin) & model comparison where $p_a = 1/2$. The forward probability is

$$P(s|H_o) = \frac{1}{2^F}$$

Question: Given a string of F observations, what comparison can we make between the biased coin model and the fair coin model, H_o vs H_1?
The Hypothesis space is now $\{H_o, H_1\}$ where only models are under consideration. Using Baye's theorem again

$$P(H_o|s, F) = \frac{P(s|F, H_o)P(H_o)}{P(s|F)}$$

and

$$P(H_1|s, F) = \frac{P(s|F, H_1)P(H_1)}{P(s|F)}$$

where $P(s|F) = \sum_{H \in \{H_o, H_1\}} P(s|F, H)P(H)$. looking at the ratio of the two probabilities

$$\frac{P(H_1|s, F)}{P(H_0|s, F)} = \frac{P(s|F, H_o)}{P(s|F, H_1)} \frac{P(H_1)}{P(H_0)}$$

where the first fraction is what the data told us, and the second fraction is what we know before (prior).

Lecture 2/1/24

Last time: We discussed the zero-parameter model H_o (fair coin) and the one-parameter model H_1 (biased coin). We used Baye's theorem to compare the two models to find the ratio of the two probabilities

$$\mathcal{R} = \frac{P(H_1|s,F)}{P(H_o|s,F)} = \frac{P(H_1)}{P(H_o)} \frac{P(s|F,H_o)}{P(s|F,H_1)}$$

where we set no a priori model (prior) preference, so $P(H_1) = P(H_o) = 1/2$. So the ratio is

$$\mathcal{R} = \frac{P(s|F,H_1)}{P(s|F,H_o)} = \frac{\frac{F_a!F_b!}{(F_a+F_b+1)!}}{\frac{1}{2^F}} = \frac{2^F F_a!F_b!}{(F+1)!}$$

what does this plot look like? As the number of tosses goes to infinity, this ratio will go to the truth! Simulation is shown by Figure 3.1. where the the bent coin $p_a = 0.9$ probability goes to infinity as well

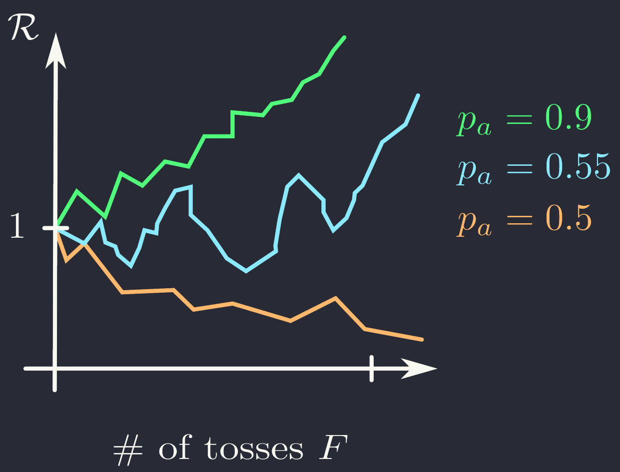


Figure 3.1: Ratio of the two probabilities as a function of the number of tosses

as the slightly biased coin (but at a slower pace) and the fair coin goes to zero. We know this from the probability

$$P(s|F,H_o) = \int_0^1 P(s|p_a,F,H_1)P(p_a|F,H_1)dp_a$$

NOTE: There exists a p_a that fits data better than H_o, but this evidence term includes averaging over p_a
Bayes theorem in the context of model comparison

$$\text{bayes} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}$$

TAKEHOME: Bayesian model comparison naturally includes Occam's Razor!

2.4 P-values? Why not just use p-values? e.g.

$$F = 250 \qquad F_a = 141, F_b = 109$$

Do these data suggest that the coin is biased?

P-value: Probability to get data as extreme or move, assuming the null hypothesis is true.

- Null hypothesis: Coin is fair (H_o)
- Our hypothesis: Coin is biased (H_1)
- mean $= F/2$
- $\sigma = \sqrt{F}/2$
- Our observation: $\frac{F_a - F/2}{\sqrt{F}/2} = 2.02\sigma$
- p-value $= 0.0497 < 0.05$!!!!

Google "a small p-value (< 0.05) indicates strong evidence against the null hypothesis so you reject it"

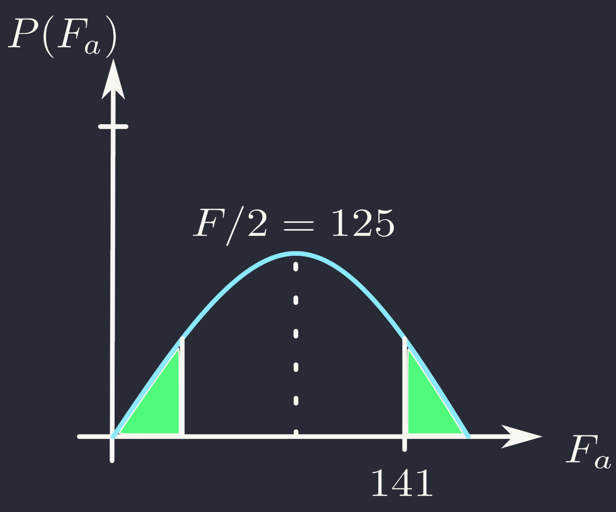


Figure 3.2: Finding p-value based on the Gaussian distribution

From sterling approximation

$$ln(k!) \approx kln(k) - k + \dots$$

With uniform prior on p_a

$$\mathcal{R} = \frac{2^2 50141!109!}{251!} = 0.61$$

if anything, there is weak evidence *against* coin being biased.

Non-uniform priors? For a reasonable family of priors, across the entire set of priors, strongest evidence for bias is 2.5 : 1 (From Mackay) This differs from the p-value which is 20 : 1.

4 Chapter 3: Maximum Likelihood *Approximation*

(Ch 22 Mackay)

GOAL: Connect to the stat you may have seen before. Going back to Example 4 (Urns and more urns)

- Unkown $u*$ selected at random
- 10 draws (with replacement): 3 black

- $P(\text{next draw} = \text{black}) = ?$
- Most likeley $u : 3 \rightarrow$ predicts 0.3
- Correct answer: predicts 0.33

but the two numbers are kinda similar...
NOTE: Bayesian model comparison, not model selection, but complete enumeration of hypothesese (integration over hyp space) is computationally expensive (especially in high dimensions)
e.g. Comparing 2 models:

- 1 Gaussian: 2 parameters μ, σ
- 2 Gaussian ($a_1 G_1 + a_2 G_2$): 5 parameters $\mu_1, \sigma_1, \mu_2, \sigma_2, a_1/a_2$

This problem of an increasing number of parameters motivates *Max likelihood (ML) approximation*: instead of enumeration, focus on *1* hypothesis that maximized the likelihood function.

Max Likelihood Estimation (MLE)

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

instead of [assuming prior $\rightarrow$ compute posterior $\rightarrow$ integrate over hyp space] we just [compute the likelihoood unction $\rightarrow$ maximize it] (MLE).

3.1 A single Gaussian

- Data: $\{x_n\} \quad n = 1, \ldots, N$
- model: these observations were sampled from a gaussian with probability

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

  where we have 2 parameters μ, σ to determine.

Log likelihood (multiplying likelihoods is hard, adding log likelihoods is easier)

$$\begin{aligned} \ln P(\{x_n\}|\mu, \sigma) &= \sum_{n=1}^{N} \left(-\ln \sqrt{2\pi\sigma^2} - \frac{(x_n - \mu)^2}{2\sigma^2} \right) \\ &= -N \ln \sqrt{2\pi\sigma^2} - \frac{N}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu)^2 \end{aligned}$$

Sufficient statistics: Denote

$$\begin{aligned} \hat{x} &\equiv \sum_n \frac{x_n}{N} \quad \text{empirical mean} \\ S &= \sum_n (x_n - \hat{x})^2 \quad \text{sum of square deviations} \end{aligned}$$

These two numbers refer to the sufficient statistics. From these we get the log likelihood

$$\ln P = -N \ln \sqrt{2\pi\sigma^2} - \frac{N(\mu - \hat{x})^2 + S}{2\sigma^2}$$

Thus the max likelihood estimate of μ, σ are

$$\begin{aligned} \mu_{ML} &= \hat{x} \\ \sigma_{ML} &= \sqrt{\frac{S}{N}} = \sqrt{\frac{\sum_n (x_n - \hat{x})^2}{N}} \end{aligned}$$

If σ is known, then $P(\mu)$ is a Gaussian we know that $\sigma/\sqrt{N}$ is the width of the likelihood (error bars)

Homework 1

Due 1/30 12pm

---

1. (a) For case 1.1 the marginal probabilty is

$$P(x=0)=0.2, \quad P(x=1)=0.8$$
$$P(y=0)=0.6, \quad P(y=1)=0.4$$

For 1.2

$$P(x=0)=0.4, \quad P(x=1)=0.6$$
$$P(y=0)=0.6, \quad P(y=1)=0.4$$

(b) 1.1

$$P(x=0|y=0)=1/5, \quad P(x=1|y=0)=4/5$$
$$P(x=0|y=1)=1/5, \quad P(x=1|y=1)=4/5$$

1.2

$$P(x=0|y=0)=1/3, \quad P(x=1|y=0)=2/3$$
$$P(x=0|y=1)=3/7, \quad P(x=1|y=1)=4/7$$

(c) Variables x and y are independent iff $P(x,y)=P(x)P(y)$. For 1.1

$$P(x=0,y=0)=0.12 \quad \text{and} \quad P(x=0)P(y=0)=0.2(0.6)=0.12$$
$$P(x=0,y=1)=P(x=0)P(y=1)=0.2(0.4)=0.08\ldots$$
$$P(x,y)=P(x)P(y)$$

So x and y are independent for 1.1. You can also see that the condition of y does not change the marginal probability of x. For 1.2 there is a simple counterexample

$$P(x=0,y=0)=0.1 \quad \text{and} \quad P(x=0)P(y=0)=0.4(0.3)=0.12$$
$$P(x=0,y=0)\neq P(x)P(y)$$

So x and y are not independent (dependent) for 1.2. You can also see that the conditional probability is not the same as the marginal probability for both cases.

2. For two random variables x and y to be independent, it must be true that

$$P(x,y)=P(x)P(y) \tag{1}$$

and from the definition of conditional probability

$$P(x|y=y_o)=\frac{P(x,y=y_o)}{P(y=y_o)}$$

substituting (1) into the joint probability

$$P(x|y=y_o)=\frac{P(x)P(y=y_o)}{P(y=y_o)}$$
$$P(x|y=y_o)=P(x)$$

3. (a) Since the two thrown dice are independent, the fair dice has 36 possible outcomes $A_{xy} = \{(1,1),(1,2),\ldots,(6,6)\}$ with equal probability

$$P(x,y) = P(x)P(y) = \frac{1}{6}\cdot\frac{1}{6} = \frac{1}{36}$$

The probability distribution of the sum of the two dice $P(S)$ is

$$P(S) = \begin{cases} 1/36 & S = 2, 12 \\ 2/36 & S = 3, 11 \\ 3/36 & S = 4, 10 \\ 4/36 & S = 5, 9 \\ 5/36 & S = 6, 8 \\ 6/36 & S = 7 \end{cases}$$

where $S = x + y$. For the absolute difference of the two dice $D = |x - y|$

$$P(D) = \begin{cases} 2/36 & D = 5 \\ 4/36 & D = 4 \\ 6/36 & D = 3 \\ 8/36 & D = 2 \\ 10/36 & D = 1 \\ 6/36 & D = 0 \end{cases}$$

for the difference $D = 0$ there are 6 possible outcomes $(1,1),(2,2),\ldots,(6,6)$, for $D = 1$ there are 10 possible outcomes $(1,2),(2,1),(2,3),(3,2),\ldots,(5,6),(6,5)$, and so on.

(b) For 100 dice, the probability distribution of the sum of the dice $P(S)$ would be roughly

$$P(S) = \begin{cases} 1/6^{100} & S = 100, 600 \\ 100/6^{100} & S = 101, 599 \\ 5050/6^{100} & S = 102, 598 \\ \vdots & \vdots \\ 1.52\times 10^{76}/6^{100} & S = 350 \end{cases}$$

first we find the mean of 1 independent dice roll (μ_1)':

$$\mu_1 = \sum_x P(x)x = \frac{1}{6}\sum_{x=1}^{6} x = \frac{21}{6} = 3.5$$

because the mean of N independent dice rolls is the sum of the means of each dice roll

$$\mu_N = \sum_{i=1}^{N} \mu_i$$

thus the mean of 100 dice rolls is

$$\mu_{100} = 100 \cdot \mu_1 = \boxed{350}$$

To find the Standard Deviation we first find the variance of 1 independent dice roll (σ_1^2):

$$\begin{aligned} \mathrm{Var}[x] = \mathrm{E}[(x - \mathrm{E}[x])^2] &= \mathrm{E}[x^2 - 2x\mathrm{E}[x] + \mathrm{E}[x]^2] \\ &= \mathrm{E}[x^2] - 2\mathrm{E}[x]^2 + \mathrm{E}[x]^2\mathrm{E}[1] = \mathrm{E}[x^2] - \mathrm{E}[x]^2 \end{aligned}$$

or in summation notation

$$\sigma_1^2 = \sum_x P(x)(x - \mu_1)^2 = \frac{1}{6}\sum_{x=1}^{6}(x - 3.5)^2 = \frac{17.5}{6} = 2.9167$$

for 2 independent variables x and y

$$\begin{aligned}\mathrm{Var}[x+y] &= \mathrm{E}[((x+y)-\mathrm{E}[x+y])^2] \\ &= \mathrm{E}[(x-\mathrm{E}[x])+(y-\mathrm{E}[y])^2] \\ &= \mathrm{E}[(x-\mathrm{E}[x])^2+(y-\mathrm{E}[y])^2+2(x-\mathrm{E}[x])(y-\mathrm{E}[y])] \\ &= \mathrm{E}[(x-\mathrm{E}[x])^2]+\mathrm{E}[(y-\mathrm{E}[y])^2]+2\mathrm{E}[(x-\mathrm{E}[x])(y-\mathrm{E}[y])] \\ &= \mathrm{Var}[x]+\mathrm{Var}[y]+2\mathrm{E}[(x-\mathrm{E}[x])(y-\mathrm{E}[y])]\end{aligned}$$

where the third term is

$$\begin{aligned}\mathrm{E}[(x-\mathrm{E}[x])(y-\mathrm{E}[y])] &= \mathrm{E}[xy-x\mathrm{E}[y]-y\mathrm{E}[x]+\mathrm{E}[x]\mathrm{E}[y]] \\ &= \mathrm{E}[xy]-\mathrm{E}[x]\mathrm{E}[y]\end{aligned}$$

and for independent variables x and y the third term is zero. Thus the variance of the sum of N independent dice rolls is

$$\mathrm{Var}[N] = N\sigma_1^2 = 100 \cdot 2.9167 = 291.67$$

and the standard deviation is

$$\sigma_N = \sqrt{\mathrm{Var}[N]} = \sqrt{291.67} = \boxed{17.08}$$

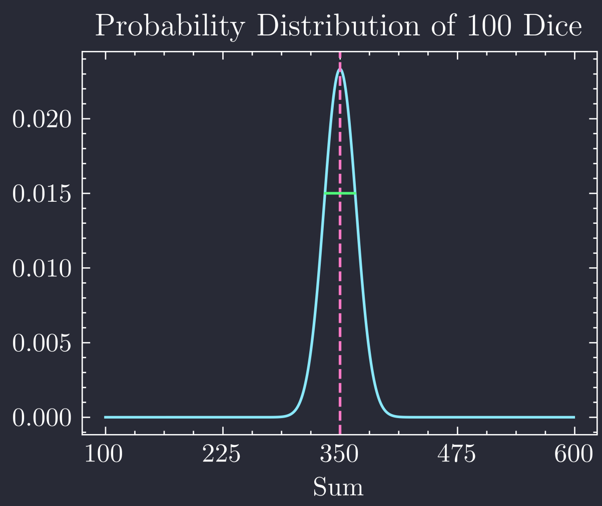


Figure 1.1: Probability distribution of the sum of 100 dice rolls. The mean is 350 and the standard deviation is ≈ 17.

The sketch of probability distribution of the sum of 100 dice rolls is shown in Figure 1.1.

4. (a) Assuming that there is an equal likelihood of the order of the age of the three brothers being any of the 6 possible permutations:

$$\mathrm{Age}_{A,B,F} = \{ABF, AFB, BAF, BFA, FAB, FBA\}$$

where we denote the first element of a permutation as the oldest brother and the last element as the youngest brother.
The probability that Fred (F) is older than Bob (B) is $\boxed{1/2}$ from both the 3 possible permutations or by realizing that there are only two equal outcomes when looking at only the age of Fred vs Bob.
(b) Given that Fred is older than Alex (A), we can eliminate the 3 permutations where Alex is older. Thus the probability that Fred is older than Bob is $\boxed{2/3}$.

5. (a) Given that the probability of choosing a black ball from an urn is $f_B = \frac{B}{K}$, the probability distribution of choosing n_B black balls from N draws is

$$\boxed{P(n_B|N, f_B) = \binom{N}{n_B} f_B^{n_B}(1-f_B)^{N-n_B}}$$

(b) Finding the mean and standard deviation of n_B is quite similar to Problem 3. Since each draw is independent, the mean of n_B is the sum of the means of each draw! In the case of drawing one ball has a binary outcome

$$n_B(N=1) = \begin{cases} 1 & \text{black ball with probability } f_b \\ 0 & \text{white ball } (1-f_B) \end{cases}$$

thus the mean of n_B for one draw is

$$\mu_1 = 1(f_B) + 0(1-f_B) = f_B$$

and the variance is

$$\sigma_1^2 = (1-f_B)^2 f_B + (0-f_B)^2(1-f_B) = f_B(1-f_B)$$

for N draws the means add up to

$$\boxed{\mu = N f_B}$$

and the variances add only due to the independent nature of the draws

$$\sigma^2 = N f_B(1-f_B)$$

thus the standard deviation is

$$\boxed{\sigma = \sqrt{N f_B(1-f_B)}}$$

For $K = 20$, and $B = K$; $f_B = 5/20 = 0.25$. And for $N = 5$ we have the ratio

$$\frac{\sigma}{\mu} = \frac{\sqrt{5 \cdot 0.25 \cdot 0.75}}{5 \cdot 0.25} = \frac{\sqrt{15}}{5} \approx \boxed{0.77}$$

and for $N = 20$

$$\frac{\sigma}{\mu} = \frac{\sqrt{1000 \cdot 0.25 \cdot 0.75}}{1000 \cdot 0.25} = \frac{\sqrt{30}}{100} \approx \boxed{0.05}$$

6. (a) Dividing the time period T in to M intervals where each interval has a probability $r\,\mathrm{d}t$ or $\mathrm{d}t = T/M$. The probability of no events occuring in time T is

$$\lim_{M\to\infty} (1 - r\,\mathrm{d}t)^M = \lim_{M\to\infty} \left(1 - \frac{rT}{M}\right)^M = \boxed{e^{-rT}}$$

(b) For $n_T = x$ events occuring in M this is similar to the binomial distribution as $M \to \infty$.

$$\lim_{M\to\infty} \frac{M!}{x!(M-x)!} \left(\frac{rT}{M}\right)^x \left(1 - \frac{rT}{M}\right)^{M-x}$$

canceling out some terms...

$$\frac{M!}{(M-x)!} = \frac{M(M-1)\cdots(M-x+1)(M-x)!}{(M-x)!} = M(M-1)\cdots(M-x+1)$$

and now the facorial has x terms, so we can write it as

$$\begin{aligned}\frac{M(M-1)\cdots(M-x+1)}{M^x} &= \frac{M}{M}\frac{M-1}{M}\cdots\frac{M-x+1}{M} \\ &= 1\cdot\left(1-\frac{1}{M}\right)\cdots\left(1-\frac{x-1}{M}\right)\end{aligned}$$

and as $M\to\infty$ the terms in the product go to 1, so the product goes to 1. Thus we are left with

$$\lim_{M\to\infty}\frac{(rT)^x}{x!}\left(1-\frac{rT}{M}\right)^{M-x} = \lim_{M\to\infty}\frac{(rT)^x}{x!}\left(1-\frac{rT}{M}\right)^{M}\left(1-\frac{rT}{M}\right)^{-x}$$

the second term is the limit of the exponential function

$$\lim_{M\to\infty}\left(1-\frac{rT}{M}\right)^{M} = e^{-rT}$$

and the third term tends to 1 as $M\to\infty$. Thus the probability of x events occuring in time T is

$$\boxed{P(x) = \frac{(rT)^x}{x!}e^{-rT}}$$

where $x = n_T$ for the sake of brevity in notation.
(c) The mean of n_T is

$$\begin{aligned}\mu &= \sum_{x=0}^{\infty} xP(x) \\ &= \sum_{x=0}^{\infty} x\frac{(rT)^x}{x!}e^{-rT} \\ &= e^{-rT}\sum_{x=1}^{\infty} x\frac{(rT)(rT)^{x-1}}{x(x-1)!} \\ &= e^{-rT}(rT)\sum_{x=1}^{\infty}\frac{(rT)^{x-1}}{(x-1)!}\end{aligned}$$

the the first term of the sum is zero which is why the sum starts at $x = 1$. The sum is also the Taylor series expansion of e^{rT} if we let $n = x-1$ so

$$\sum_{x=1}\frac{(rT)^{x-1}}{(x-1)!} = \sum_{n=0}\frac{(rT)^n}{n!} = e^{rT}$$

Therefore the mean of n_T is

$$\mu = (rT)e^{-rT}e^{rT} = rT$$

The variance of n_T is

$$\sigma^2 = \mathrm{E}[x^2] - \mathrm{E}[x]^2$$

the first term is solved similarly to the mean

$$\begin{aligned}
\mathrm{E}[x^2] &= \sum_{x=0}^{\infty} x^2 P(x) \\
&= \sum_{x=0}^{\infty} x^2 \frac{(rT)^x}{x!} e^{-rT} \\
&= e^{-rT} \sum_{x=1}^{\infty} x^2 \frac{(rT)(rT)^{x-1}}{x(x-1)!} \\
&= (rT)e^{-rT} \sum_{x=1}^{\infty} x \frac{(rT)^{x-1}}{(x-1)!} \\
&= (rT)e^{-rT} \left[\sum_{x=1}^{\infty} (x-1) \frac{(rT)^{x-1}}{(x-1)!} + \sum_{x=1}^{\infty} \frac{(rT)^{x-1}}{(x-1)!} \right] \qquad x = [(x-1)+1] \\
&= (rT)e^{-rT} \left[(rT) \sum_{x=2}^{\infty} \frac{(rT)^{x-2}}{(x-2)!} + \sum_{n=0}^{\infty} \frac{(rT)^n}{n!} \right] \qquad n = x-1 \\
&= (rT)e^{-rT} \left[(rT) \sum_{l=0}^{\infty} \frac{(rT)^l}{l!} + \sum_{n=0}^{\infty} \frac{(rT)^n}{n!} \right] \qquad l = x-2 \\
&= (rT)e^{-rT} \left[(rT)e^{rT} + e^{rT} \right] \\
&= (rT)^2 e^{-rT} e^{rT} + (rT)e^{-rT} e^{rT} \\
&= (rT)^2 + rT
\end{aligned}$$

and the variance is

$$\sigma^2 = (rT)^2 + rT - (rT)^2 = rT$$

Therefore the mean and standard deviation of n_T are

$$\boxed{\mu = rT \quad \text{and} \quad \sigma = \sqrt{rT}}$$

7. Using Bayes' theorem for the outcome $X = \{7, 3, 4, 2, 5, 3\}$ is

$$P(A|7,3,4,2,5,3) = P(A|X) = \frac{P(X|A)P(A)}{P(X)}$$

where the probability of choosing dice A is 1 in 3—$P(A) = 1/3$. The conditional probabilty $P(X|A)$ is the probability of rolling the outcome X given that dice A is chosen:

$$P(X|A) = \frac{1 \times 4 \times 2 \times 4 \times 2 \times 4}{20^6} = \frac{256}{20^6}$$

and the probability of rolling the outcome X is given by the sum rule

$$P(X) = P(X|A)P(A) + P(X|B)P(B) + P(X|C)P(C)$$

where $P(B) = P(C) = 1/3$ and the conditional probabilities $P(X|B)$ and $P(X|C)$ are

$$\begin{aligned}
P(X|B) = \frac{2 \times 3 \times 2 \times 2 \times 2 \times 3}{20^6} &= \frac{144}{20^6} \\
P(X|C) = \frac{2^6}{20^6} &= \frac{64}{20^6}
\end{aligned}$$

thus the probability of choosing dice A given the outcome X is

$$P(A|X) = \frac{\frac{256}{20^6} \cdot \frac{1}{3}}{\frac{256}{20^6} \cdot \frac{1}{3} + \frac{144}{20^6} \cdot \frac{1}{3} + \frac{64}{20^6} \cdot \frac{1}{3}} = \frac{256}{464} \approx \boxed{0.55}$$

with the knowledge that terms cancel out, the probability of the die being B is

$$P(B|X) = \frac{144}{464} \approx \boxed{0.31}$$

and the probability of the die being C is

$$P(C|X) = \frac{64}{464} \approx \boxed{0.14}$$

8. (a) Given that the bus arrives on average every 5 minutes, the average wait time is 5 minutes. And the bus that just left Sally would have left an average of 5 minutes ago. From the code, taking the mean value of the wait times is also ≈ 5 minutes.
(b) Therefore the average time between two buses is the sum in the time Sally is waiting for the bus and how long the missed bus has been gone: 10 minutes.
(c) The paradox is that 'we' think that after waiting for 5 minutes the bus will arrive, but the average time between buses is 10 minutes, so we are waiting longer than we expect to intuitively. This is because the conditional probability of Sally getting to the bus stop where the interval between buses is less than 5 minutes given that she has waited for a time t is less as time goes on. And the probability that Sally arrived at the bus stop where the interval between buses is more than 5 minutes given that she has waited for a time t is more as time goes on.
(d)

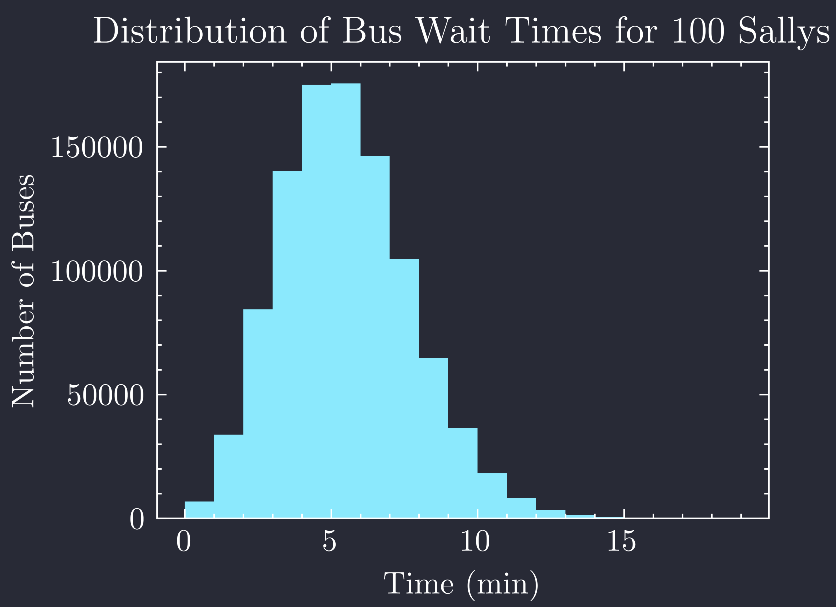


Figure 1.2: Mean of 5.00 min and Time between buses of 10.00 min.

PYTHON CODE BELOW

hw1_python

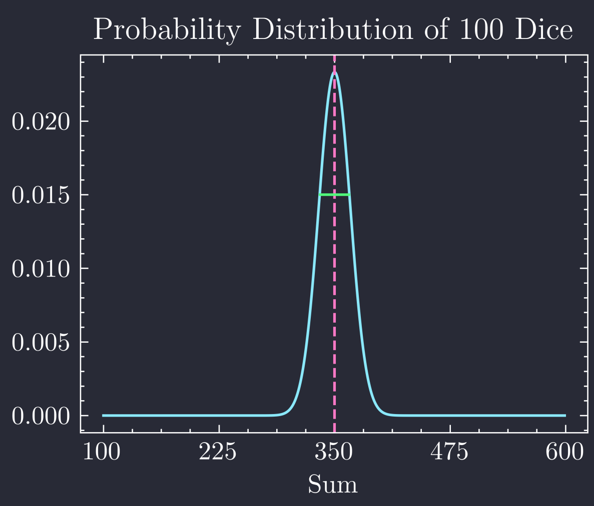
January 30, 2024

[ ]:
```
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
import matplotlib as mpl
import scienceplots

# Science plot package + Dracula theme
plt.style.use(['science', 'dark_background'])
plt.rcParams['axes.facecolor'] = '#282a36'
plt.rcParams['figure.facecolor'] = '#282a36'
colorcycle = ['#8be9fd', '#ff79c6', '#50fa7b', '#bd93f9', '#ffb86c', '#ff5555',␣
↪'#f1fa8c',
'#6272a4']
plt.rcParams['axes.prop_cycle'] = mpl.cycler(color=colorcycle)
white = '#f8f8f2' # foreground

# change dpi
plt.rcParams['figure.dpi'] = 1024
```

[ ]:
```
# highly optimized function to count the number of ways to get a sum of s with␣
↪d dice
def count_ways_to_sum(sum_target, num_dice):
    # Initialize a 2D array to store results of subproblems
    dp = [[0] * (sum_target + 1) for _ in range(num_dice + 1)]

    # Base case: there is one way to get a sum of 0 (no dice)
    dp[0][0] = 1

    # Fill the dp table using the convolution formula
    for d in range(1, num_dice + 1):
        for s in range(1, sum_target + 1):
            for k in range(1, 7):
                if s - k >= 0:
                    dp[d][s] += dp[d-1][s-k]

    return dp[num_dice][sum_target]
```

```python
# Example usage:
sum_100_ways = count_ways_to_sum(100, 100)
sum_101_ways = count_ways_to_sum(101, 100)
sum_102_ways = count_ways_to_sum(102, 100)
sum_350_ways = count_ways_to_sum(350, 100)

print(f"Number of ways to get a sum of 100 with 100 dice: {sum_100_ways}")
print(f"Number of ways to get a sum of 101 with 100 dice: {sum_101_ways}")
print(f"Number of ways to get a sum of 102 with 100 dice: {sum_102_ways}")
print(f"Number of ways to get a sum of 350 with 100 dice: {sum_350_ways:.2e}")

# for loop to get a function of sum to combinations
sums = np.arange(100, 600, 1)
ways = []
for i in sums:
    ways.append(count_ways_to_sum(i, 100) / 6**100)

# plotting
plt.figure()
plt.plot(sums, ways)
plt.xlabel('Sum')
plt.title('Probability Distribution of 100 Dice')

# Add x-axis label at 350 as "mean"
plt.axvline(x=350, color=colorcycle[1], linestyle='--')

# Add tick label at 500/4 intervals
plt.xticks(np.arange(100, 601, 125))

# Add standard deviation at x= 350 +/- 17 as horizontal line
plt.hlines(y=0.015, xmin=350-17, xmax=350+17, color=colorcycle[2])

plt.show()

print (f"Probability of getting a sum of 350 with 100 dice: {sum_350_ways /
 ↪6**100:.2e}")
# sum up ways
print (f"Probability distribution adds to 1: {sum(ways):.2e}")
```

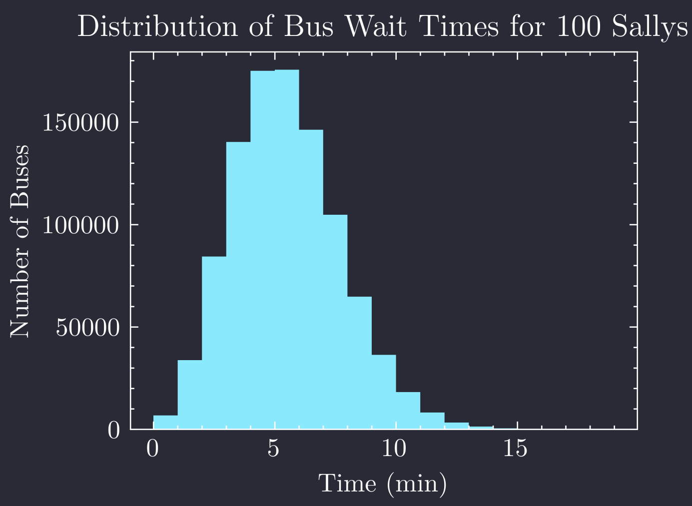
```
Number of ways to get a sum of 100 with 100 dice: 1
Number of ways to get a sum of 101 with 100 dice: 100
Number of ways to get a sum of 102 with 100 dice: 5050
Number of ways to get a sum of 350 with 100 dice: 1.52e+76
```




```
Probability of getting a sum of 350 with 100 dice: 2.33e-02
Probability distribution adds to 1: 1.00e+00
```

[ ]:
```python
# 481 Problem 8d
# Simulation of 10000 buses and 100 Sally's

# constants
N = 10000 # number of buses
t_avg = 5 # average time between buses

# using poisson distribution old way
t_poisson = np.random.poisson(t_avg, N)

# simulating 100 Sally's
for i in range(99):
    t_poisson = np.concatenate((t_poisson, np.random.poisson(t_avg, N)))

# checking if it is 100 Sally's
print(len(t_poisson) / 10000)

# plotting
plt.figure()
plt.hist(t_poisson, bins=np.max(t_poisson))
plt.xlabel('Time (min)')
plt.ylabel('Number of Buses')
plt.title('Distribution of Bus Wait Times for 100 Sallys')
plt.show()
```

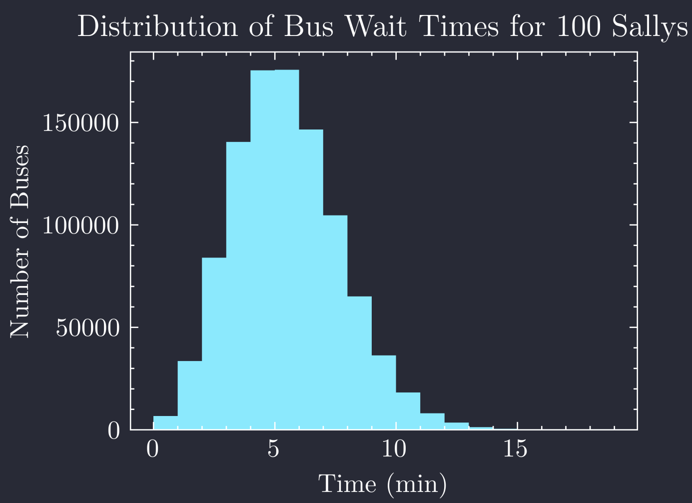
```
100.0
```



[ ]:
```python
# implementing hard code
# random number generator
rand = np.random.default_rng(seed=42)

# Probability of 1 bus given 5 min avg
p = np.exp(-5)

# wait time for 1 bus
def wait_time():
    time = 0
    prod = 1.0
    while True:
        U = rand.random()
        prod *= U
        if prod > p:
            time += 1
        else:
            return time

def wait_time_100():
    times = []
    for i in range(100):
        times.append(wait_time())
    return times

# simulating 10000 buses
times = []
for i in range(10000):
    times += wait_time_100()
```

```
print(np.mean(times))

# plotting
plt.figure()
plt.hist(times, bins=np.max(times))
plt.xlabel('Time (min)')
plt.ylabel('Number of Buses')
plt.title('Distribution of Bus Wait Times for 100 Sallys')
plt.show()
```

5.001477



[ ]:

```
# time of bus that just passed
last_bus_times = []
for i in range(1, len(times)):
    if times[i] == 0:
        last_bus_times.append(times[i] - times[i - 1])

print(np.mean(last_bus_times))

# time in between buses
between_bus_times = []
for i in range(1, len(times)):
    between_bus_times.append(times[i] + times[i-1])

print(np.mean(between_bus_times))
```

-5.012881255552266
10.002945002945003

Homework 2

Due 2/6 12pm

1.

$$P(r|\lambda) = \exp(-\lambda)\frac{\lambda^r}{r!}$$

(a) Taking the log of the likelihood function:

$$L(\lambda) = \ln P(r|\lambda) = -\lambda + r\ln\lambda - \ln r!$$

finding the maximum by taking the derivative with respect to λ and setting it to zero:

$$\frac{dL}{d\lambda} = -1 + \frac{r}{\lambda} = 0 \implies \hat{\lambda} = r$$

so the maximum likelihood estimate for λ is $\hat{\lambda} = r$.
(b) Given the derivative with respect to the function $\ln\lambda$:

$$\frac{\mathrm{d}}{\mathrm{d}(\ln\lambda)}u^n = nu^n, \qquad \frac{\mathrm{d}}{\mathrm{d}(\ln\lambda)}\ln\lambda = 1$$

we can find the curvature of the log likelihood function:

$$\frac{\mathrm{d}}{\mathrm{d}(\ln\lambda)}L(\lambda) = -\lambda + r = 0 \implies \hat{\lambda} = r$$
$$\frac{\mathrm{d}^2}{\mathrm{d}(\ln\lambda)^2}L(\lambda) = -\lambda = k$$

For a normal distribution with width σ, the curvature is $k = -1/\sigma^2$. So the width is approximately

$$\sigma \propto \frac{1}{\sqrt{-k}} = \frac{1}{\sqrt{\lambda}}$$

and the 95% confidence interval at the MLE is approximately

$$\hat{\lambda} \pm 2\sigma = r \pm \frac{2}{\sqrt{\hat{\lambda}}}$$

(c) Given the new Poisson distribution

$$P(r|\lambda) = \exp(-(\lambda + b))\frac{(\lambda + b)^r}{r!}$$

the log likelihood function is

$$L(\lambda) = -(\lambda + b) + r\ln(\lambda + b) - \ln r!$$

and the maximum likelihood estimate for λ is

$$\frac{dL}{d\lambda} = -1 + \frac{r}{\lambda + b} = 0 \implies \hat{\lambda} = r - b$$

the value $\hat{\lambda} = 9 - 13 = -4$ is not physically meaningful, so the MLE will be the lowest possible value for λ which is $\hat{\lambda} = 0$. From this we can infer that the remote star is very dim. The Bayesian posterior distribution for λ is

$$P(\lambda|r) = \frac{P(r|\lambda)P(\lambda)}{P(r)}$$

and sketched in the figure below

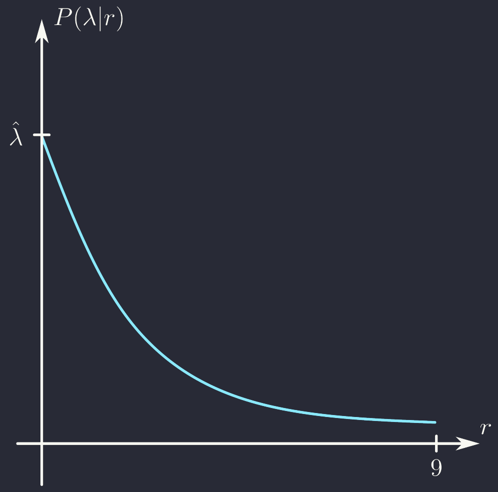


Figure 1.3: The posterior distribution for λ given r

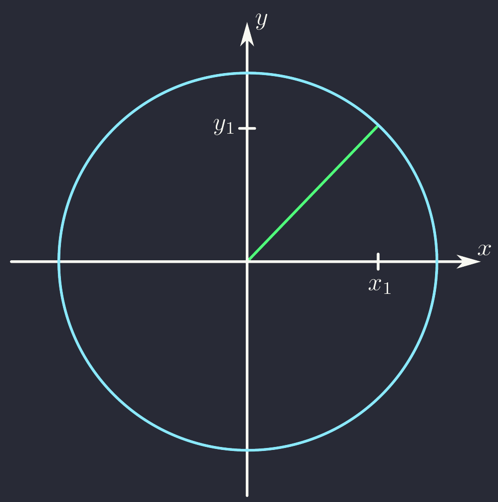


Figure 1.4: Segment of Gaussian distribution at $\{x_1, y_1\}$

2. (a) From the geometric picture as shown in Figure 1.4, the segment of the Gaussian is a circle with radius $\rho = \sqrt{x_1^2 + y_1^2}$ and the circumference is $2\pi\rho$ which directly relates to the extra factor of ρ and canceling the 2π in the denominator. This is also related to the Jacobian when transforming from Cartesian to polar coordinates when computing the integral. Using the integral of a 1D Gaussian:

$$\int_{-\infty}^{\infty} \exp(-ax^2)\,\mathrm{d}x = \sqrt{\frac{\pi}{a}}$$

Verifying that the integral is normalized:

$$\begin{aligned}
\frac{1}{2\pi\sigma^2}\iint_{-\infty}^{\infty} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)\mathrm{d}x\,\mathrm{d}y &= \frac{1}{\sigma^2}\int_0^{\infty} \exp\left(-\frac{\rho^2}{2\sigma^2}\right)\rho\,\mathrm{d}\rho \\
\frac{1}{2\pi\sigma^2}\int_{-\infty}^{\infty} \exp\left(-\frac{x^2}{2\sigma^2}\right)\mathrm{d}x\int_{-\infty}^{\infty} \exp\left(-\frac{y^2}{2\sigma^2}\right)\mathrm{d}y &= \frac{1}{\sigma^2}\int_0^{\infty} \exp\left(-\frac{\rho^2}{2\sigma^2}\right)\rho\,\mathrm{d}\rho \\
\frac{1}{2\pi\sigma^2}\sqrt{2\pi\sigma^2}\sqrt{2\pi\sigma^2} &= \frac{1}{\sigma^2}\int_0^{\infty} \sigma^2\exp(-u)\,\mathrm{d}u \\
\frac{2\pi\sigma^2}{2\pi\sigma^2} &= \left[-e^{-u}\right]\Big|_0^{\infty} = 1
\end{aligned}$$

so both integrals are normalized.

(b)

$$P(\rho) = \frac{\rho}{\sigma_w^k} \exp\left(-\frac{\rho^2}{2\sigma_w^2}\right)$$

The sketch for the distribution of $P(\rho)$ is shown in Figure 1.5.

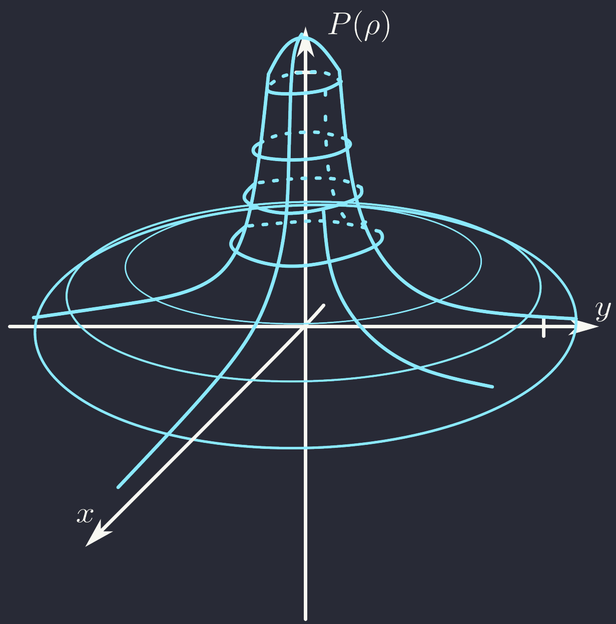


Figure 1.5: The distribution of ρ for $k = 1000$

(c) The standard deviation is

$$\sigma_w = \sqrt{\frac{\sum(w-\mu)^2}{k}}$$

and because the distribution is centered at the origin, the mean is $\mu = 0$, so

$$\sigma_w = \sqrt{\frac{\rho^2}{k}}$$

Since most of the probability mass lies around ρ or equivalently a radius of the thin shell where $\rho = r = \sigma_w\sqrt{k}$, and the thickness of the shell is equivalent to the standard deviation of the gaussian $= r/\sqrt{k}$.
(d) Taking the ratio of the probability density from the origin to a point $\rho = \omega_w\sqrt{k}$ away:

$$\frac{P(0)}{P(\sigma_w\sqrt{k})} = \exp\left\{\frac{\sigma_w^2 k}{2\sigma_w^2}\right\} = \exp\left\{\frac{k}{2}\right\}$$

(e) For a shell to contain 95% of the probability mass, $\sigma_w = 2$ and thus the radius and thickness of the shell are

$$r = \sigma_w\sqrt{k} = 2\sqrt{1000} = 63.25, \qquad \frac{r}{\sqrt{1000}} = 2$$

and the probability density is $\exp\{500\} \approx 10^{217}$ larger at the origin than at the edge of the shell.
(f) For a 1% difference in σ_w at the origin, the radius term is at zero so the exponents are 1, so the ratio of the probability densities is

$$\frac{(1.01\sigma_w)^k}{\sigma_w^k} = 1.01^k \approx 20959.$$

(g) Because of the large amount of parameters yet a very dense probability mass in a small region, the weight of one parameter is too small for the MLE to represent each of the many parameters, so adding or taking out a parameter will not significantly change the MLE.

3. (a) From class

$$\mathcal{R} = \frac{\frac{F_a!F_b!}{(F+1)!}}{1/2^F} = \frac{2^F F_a!F_b!}{(F+1)!}$$

(b) Taking the log of the ratio and using Stirling's approximation as shown in the code we get the plot of the three simulations in Figure 1.6

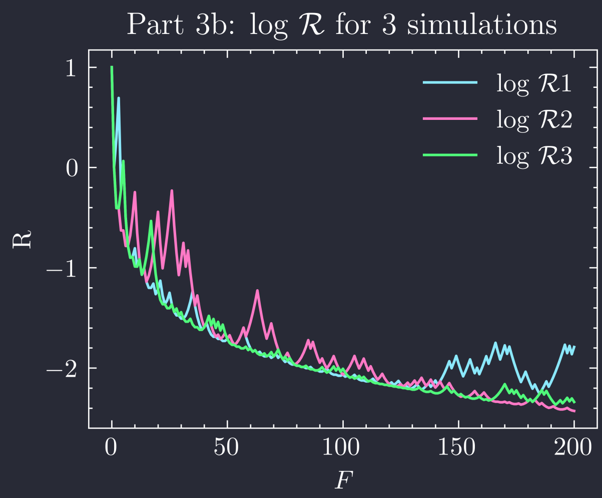


Figure 1.6: The log of the ratio of the likelihoods for the three simulations

(c) The trajectory of the two biased coin models are shown in Figure 1.7 and 1.8.

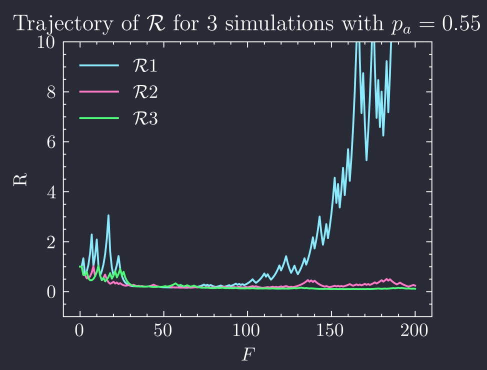


Figure 1.7: There isn't clear evidence for a bias in the first 200 flips

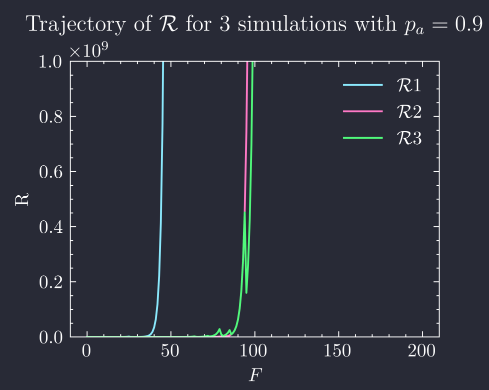


Figure 1.8: A clear bias is shown in the first 200 flips here

and the posterior distribution for the two biased coin models are shown in Figure 1.9 and 1.10.

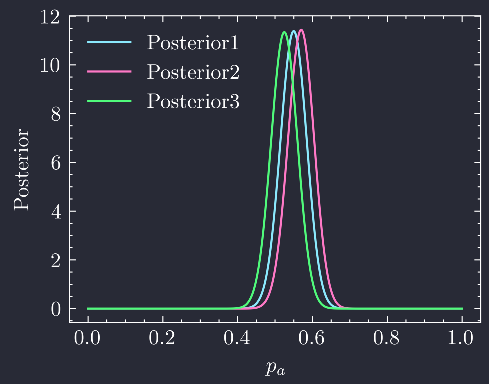


Figure 1.9: We can see that the distribution is centered around roughly $p_a = 0.55$ as expected

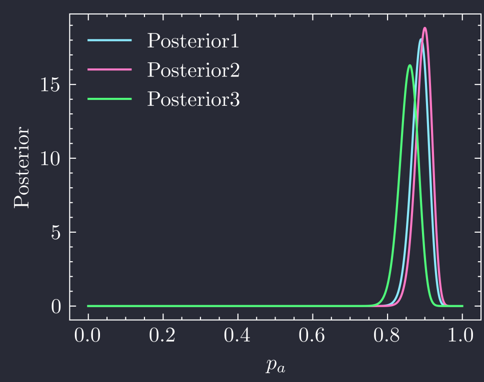


Figure 1.10: There is definitely a bias towards $p_a = 0.9$ in this distribution

(d) From the Gaussian, we can approximate the error bars as $p_a - \frac{1}{2}$, so

$$p_a - 0.5 = \frac{\sigma}{\sqrt{F}} \to F = \frac{\sigma^2}{(p_a - 0.5)^2}$$

so we can get a rough estimate of the number of flips needed to distinguish between the two models for within 2-3 standard deviations as shown in 1.11 and 1.12.

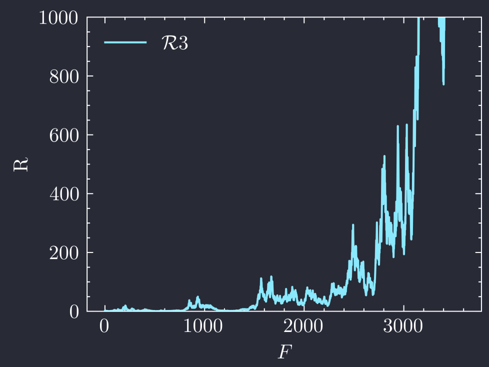


Figure 1.11: For $\sigma = 3$ the model is clearly distinguishable after ≈ 3600 flips

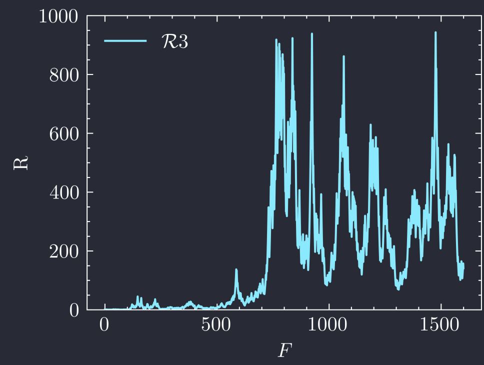


Figure 1.12: For $sigma = 2$ the model does seem to to be biased, but it is not as confident as the previous case

(e) It is suprising to find out that finding evidence against $\mathcal{H}_1$ would be slower than finding evidence for it, but in hindsight it makes sense for cases where the biased probabilities are close to the unbiased coin model since we would need an exponential number of flips as the bent coin model probabilities is closer to $p_o = 1/2$.
From the log of the ratio

$$\begin{aligned}\ln \mathcal{R} &= F \ln 2 + \ln F_a! + \ln F_b! - \ln(F+1)! \\ &= F \ln 2 + \ln(F - F_b)! + \ln(F - F_a)! - \ln(F+1)! \\ &= F \ln 2 + (F - F_b)\ln(F - F_b) - (F - F_b) + \frac{1}{2}\ln 2\pi(F - F_b)\ldots\end{aligned}$$

taking the derivative

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}F}\ln \mathcal{R} &= \ln 2 + 1 + \ln(F - F_b) - 1 + \frac{1}{2}\frac{1}{F - F_b}\ldots \\ &= \ln 2 + \ln(F_a) + \frac{1}{2F_a} + \ln(F_b) + \frac{1}{2F_b} - \ln(F+1) - \frac{1}{2(F+1)}\end{aligned}$$

The fastest $\log \mathcal{R}$ can grow is when F is very large, and the fractional terms at very large F will be negligible and the log terms grow much slower (e.g. $\ln\left(10^{20}\right) \approx 46$), so the derivative will be be approximately a constant which relates to a linear growth. The fastest $\log \mathcal{R}$ can fall is when F is small and where the $-\ln(F+1)$ term will dominate thus the growth will be approximately logarithmic.

PYTHON CODE BELOW

```python
# %%
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
import matplotlib as mpl
import scienceplots
import gmpy2 as gm

# Science plot package + Dracula theme
plt.style.use(['science', 'dark_background'])
plt.rcParams['axes.facecolor'] = '#282a36'
plt.rcParams['figure.facecolor'] = '#282a36'
colorcycle = ['#8be9fd', '#ff79c6', '#50fa7b', '#bd93f9', '#ffb86c', '#ff5555', '#f1fa8c',
'#6272a4']
plt.rcParams['axes.prop_cycle'] = mpl.cycler(color=colorcycle)
white = '#f8f8f2' # foreground

# change dpi
plt.rcParams['figure.dpi'] = 1024

# %%
# 3b
# Calculating the ratio R
# Stirling's approximation
def power(n, exp):
    result = 1
    for _ in range(exp):
        result = gm.mul(n, result)
    return result

def fact(n):
    if n < 10:
        return np.math.factorial(n)
    a = power(n, n)
    b = gm.exp(-n)
    c = gm.sqrt(2 * np.pi * n)
    return a * b * c

def logratio(F_a,F_b):
    return (F_a + F_b) * gm.log(2) + gm.log(fact(F_a)) + gm.log(fact(F_b)) - gm.log(fact
    (F_a + F_b + 1))

def flip():
    if np.random.rand() < 0.5:
        return 1
    else:
        return 0

def simulate(n):
    F_a = 0
    F_b = 0
    R = [1]
    for i in range(n):
        if flip():
            F_a += 1
        else:
            F_b += 1
        R.append(logratio(F_a,F_b))
    return R
print(fact(100))
print(2 * gm.log(fact(100)))
print(gm.log(fact(201)))
print(logratio(100, 100))
# numpy array with 1 to 200 on the x-axis
n = np.arange(0, 201)
# calculate the ratio for each n
# R = simulate(200)
```

```
# simulating 3 times and plotting them side by side
plt.figure()
for i in range(3):
    R = simulate(200)
    # Plot on the first subplot
    plt.plot(n, R, label='log $\mathcal{R}$' + str(i+1))
    plt.xlabel('$F$')
    plt.ylabel('R')
plt.title('Part 3b: log $\mathcal{R}$ for 3 simulations')
plt.legend()
plt.show()

# plot the ratio
# plt.plot(n, R, label='R')
# plt.xlabel('$F$')
# plt.ylabel('R')
# plt.legend()
# plt.show()

# %%
# for p_a = 0.55
def bflip():
    if np.random.rand() < 0.55:
        return 1
    else:
        return 0

def vbflip():
    if np.random.rand() < 0.9:
        return 1
    else:
        return 0

Fa_toss_count_55 = []
Fa_toss_count_90 = []

def ratio(F_a,F_b):
    return fact(F_a) * fact(F_b) / fact(F_a + F_b + 1) * 2 ** (F_a + F_b)

def simulate_b(n):
    F_a = 0
    F_b = 0
    R = [1]
    for i in range(n):
        if bflip():
            F_a += 1
        else:
            F_b += 1
        R = np.append(R, ratio(F_a,F_b))
    return R, F_a

print(2 ** 100 * gm.factorial(90) * gm.factorial(10) / gm.factorial(101))

print(2 ** 500 * gm.factorial(275) * gm.factorial(225) / gm.factorial(501))
def simulate_vb(n):
    F_a = 0
    F_b = 0
    R = [1]
    for i in range(n):
        if vbflip():
            F_a += 1
        else:
            F_b += 1
        R = np.append(R, ratio(F_a,F_b))
    return R, F_a

# simulating 3 times and plotting them
plt.figure()
for i in range(3):
    R, F_a = simulate_b(200)
```

```python
    Fa_toss_count_55.append(F_a)
    plt.plot(n, R, label='$\mathcal{R}$' + str(i+1))
plt.xlabel('$F$')
plt.ylabel('R')
plt.title('Trajectory of $\mathcal{R}$ for 3 simulations with $p_a = 0.55$')
plt.ylim(-1, 10)
plt.legend()

# for p_a = 0.9
plt.figure()
for i in range(3):
    R, F_a = simulate_vb(200)
    Fa_toss_count_90.append(F_a)
    plt.plot(n, R, label='$\mathcal{R}$' + str(i+1))
plt.xlabel('$F$')
plt.ylabel('R')
plt.title('Trajectory of $\mathcal{R}$ for 3 simulations with $p_a = 0.9$')
plt.legend()
plt.ylim(0, 1e9)
plt.show()

# %%
# posterior distribution for p_a = 0.55
def normalconst(F_a):
    return gm.factorial(F_a) * gm.factorial(200 - F_a) / gm.factorial(201)

def posterior(F_a, p_a):
    return p_a ** (F_a) * (1 - p_a) ** (200 - F_a) / normalconst(F_a)

for i, F_a in enumerate(Fa_toss_count_55):
    p_a = np.linspace(0, 1, 1000)
    plt.plot(p_a, posterior(F_a, p_a), label='Posterior' + str(i+1))
plt.xlabel('$p_a$')
plt.ylabel('Posterior')
plt.legend()
plt.show()

# posterior distribution for p_a = 0.9
for i, F_a in enumerate(Fa_toss_count_90):
    p_a = np.linspace(0, 1, 1000)
    plt.plot(p_a, posterior(F_a, p_a), label='Posterior' + str(i+1))
plt.xlabel('$p_a$')
plt.ylabel('Posterior')
plt.legend()
plt.show()

# %%
# back-envelope calculation for p_a close to 0.5
# for a Gaussian the width of the error bars is sigma / sqrt(n)
# the width should be less than (p_a - 0.5)
# p_a - 0.5 > sigma / sqrt(n)
# n > sigma ** 2 / (p_a - 0.5) ** 2
def backenvelope(sigma):
    return sigma ** 2 / (0.55 - 0.5) ** 2
print(backenvelope(3))
print(backenvelope(2))

l = np.arange(0, 3601)
plt.figure()
R, F_a = simulate_b(3600)
plt.plot(l, R, label='$\mathcal{R}$' + str(i+1))
plt.xlabel('$F$')
plt.ylabel('R')
plt.ylim(0, 1e3)
plt.legend()

plt.figure()
R, F_a = simulate_b(1600)
plt.plot(np.arange(0,1601), R, label='$\mathcal{R}$' + str(i+1))
plt.xlabel('$F$')
```

```
plt.ylabel('R')
plt.ylim(0, 1e3)
plt.legend()

# %%
# 2 e
print(np.e ** 500)
```