1. i) If h is the number of heads over 1000 flips and p is the probability of any one flip resulting in a head, we can formalize this result as:

$$\begin{aligned}
\frac{P(h=527\,|\,p=q\neq .5)}{P(h=527\,|\,p=.5)} &\le \max_{q\in(0,1)} \frac{P(h=527\,|\,p=q)}{P(h=527\,|\,p=.5)} \\
&\le \max_{q\in(0,1)} \frac{\binom{1000}{527}q^{527}(1-q)^{473}}{\binom{1000}{527}.5^{527}.5^{473}} \\
&\le \max_{q\in(0,1)} (2q)^{527}(2-2q)^{473} \\
&\le \left(2\frac{527}{1000}\right)^{527}\left(2-2\frac{527}{1000}\right)^{473} \\
&\le 4.30 \approx 4
\end{aligned}$$

ii) We can calculate the largest posterior probability that the coin is weighted (to a particular value of p) rather than fair using Bayes rule and our previous result

$$\begin{aligned}
\max_{q\in(0,1)} P(p=q\,|\,h=527) &= \frac{P(p=q)P(h=527\,|\,p=q)}{P(h=527)} \\
&= \frac{P(p=q)P(h=527\,|\,p=q)}{P(p=q)P(h=527\,|\,p=q)+P(p=.5)P(h=527\,|\,p=.5)} \\
&= \frac{.5\frac{P(h=527\,|\,p=q)}{P(h=527\,|\,p=.5)}P(h=527\,|\,p=.5)}{.5\frac{P(h=527\,|\,p=q)}{P(h=527\,|\,p=.5)}P(h=527\,|\,p=.5)+.5P(h=527\,|\,p=.5)} \\
&= \frac{4.30}{4.30+1} \\
&= .811
\end{aligned}$$

iii) If one started with the prior that coin was far with probability .5 and that the true portion was actually the sample mean with probability .5, then the Bayes factor in favor of the alternate hypothesis is

$$\begin{aligned}
\frac{P(p=.527\,|\,h=527)}{P(p=.5\,|\,h=527)} &= \frac{P(h=527\,|\,p=.527)P(p=.527)}{P(h=527\,|\,p=.5)P(p=.5)} \\
&= \frac{P(h=527\,|\,p=.527)}{P(h=527\,|\,p=.5)} \\
&= 4.30
\end{aligned}$$

Which is the answer from part i). If we repeat this analysis with a prior from a Beta(10,10) distribution, we get a Bayes factor of:

$$\frac{P(h=527\,|\,p=.527)P(p=.527)}{P(h=527\,|\,p=.5)P(p=.5)} = 4.30\frac{f_{10,10}(.527)}{f_{10,10}(.5)} = 4.18$$

Then, with a Beta(1,1) distribution:

$$\frac{P(h=527\,|\,p=.527)P(p=.527)}{P(h=527\,|\,p=.5)P(p=.5)} = 4.30\frac{f_{1,1}(.527)}{f_{1,1}(.5)} = 4.30$$

And finally with a Beta(.5,.5) distribution:

$$\frac{P(h=527\,|\,p=.527)P(p=.527)}{P(h=527\,|\,p=.5)P(p=.5)} = 4.30\frac{f_{.5,.5}(.527)}{f_{.5,.5}(.5)} = 4.31$$

I prefer the Beta(10,10) prior, since that is the most heavily concentrated around $p = .5$, meaning that the prior says that the coin is mostly likely fair. This fits my intuition that it is less likely that a coin is heavily biased towards heads or tails than fair or close to fair.

2. i)

$$\begin{aligned} P(X_i - 1.96 \leq \theta_i \leq X_i + 1.96 \,|\, \theta_i) &= P(\theta_i - 1.96 \leq X_i \leq \theta_i + 1.96) \\ &= \Phi(X_i + 1.96 - \theta_i) - \Phi(X_i - 1.96 - \theta_i) \\ &= .95 \end{aligned}$$

So the interval contains θ_i 95% of the time.

ii)

$$\begin{aligned} P(\theta_i \,|\, X_i, a) &= \frac{P(X_i \,|\, \theta_i, a) P(\theta_i \,|\, a)}{P(X_i \,|\, a)} \\ &= \frac{\phi(X_i - \theta_i) \frac{I_{[-a,a]}(\theta_i)}{2a}}{\int_{-a}^{a} \phi(X_i - \theta_i) \frac{1}{2a} d\theta} \\ &= \frac{\phi(X_i - \theta_i)}{\Phi(X_i + a) - \Phi(X_i - a)} I_{[-a,a]}(\theta_i) \end{aligned}$$

As $a \to \infty$, $\Phi(X_i + a) - \Phi(X_i - a) \to 1$, so $P(\theta_i \,|\, X_i, a) \to \phi(X_i - \theta_i)$. Thus

$$P(\theta_i \in x \pm 1.96 \,|\, X_i = x, a) \to \Phi(x + 1.96) - \Phi(x - 1.96) = .95$$

Making $x \pm 1.96$ a 95% credible interval in the limit.

iii) Going through the posterior calculations,

$$\begin{aligned} P(\theta_i \,|\, X_i) &= \frac{P(X_i \,|\, \theta_i) P(\theta_i)}{P(X_i)} \\ &= \frac{\phi(X_i - \theta_i)\phi(\theta_i)}{\int_{\mathbb{R}} \phi(X_i - \theta_i)\phi(\theta_i) d\theta_i} \end{aligned}$$

Which, can be simplified using the fact that

$$\begin{aligned} \phi(X_i - \theta_i)\phi(\theta_i) &= \frac{1}{2\pi} e^{-(X_i - \theta_i)^2 - \theta_i^2} \\ &= \frac{1}{2\pi} e^{-2\theta_i^2 - 2\theta_i X_i - X_i^2} \\ &= \frac{1}{2\pi} e^{-2(\theta_i - \frac{1}{2}X_i)^2 - \frac{1}{2}X_i^2} \\ &= \frac{1}{\sqrt{4\pi}} e^{-\frac{1}{2}X_i^2} \frac{1}{\sqrt{\pi}} e^{-2(\theta_i - \frac{1}{2}X_i)^2} \\ &= \phi\left(\frac{X_i}{\sqrt{2}}\right) \phi\left(\sqrt{2}\left(\theta_i - \frac{1}{2}X_i\right)\right) \end{aligned}$$

To give us that

$$\begin{aligned} P(\theta_i \,|\, X_i) &= \frac{\phi\left(\frac{X_i}{\sqrt{2}}\right) \phi\left(\sqrt{2}\left(\theta_i - \frac{1}{2}X_i\right)\right)}{\int_{\mathbb{R}} \phi\left(\frac{X_i}{\sqrt{2}}\right) \phi\left(\sqrt{2}\left(\theta_i - \frac{1}{2}X_i\right)\right) d\theta_i} \\ P(\theta_i \,|\, X_i) &= \phi\left(\sqrt{2}\left(\theta_i - \frac{1}{2}X_i\right)\right) \end{aligned}$$

Thus, $\left[-1.16 + \frac{X_i}{2}, \infty\right)$ Is a one sided 95% CI for θ_i. This is because

$$\begin{aligned} P\left(\theta_i \geq -1.16 + \frac{X_i}{2}\right) &= 1 - \Phi\left(\sqrt{2}\left(-1.16 + \frac{1}{2}X_i - \frac{1}{2}X_i\right)\right) \\ &= 1 - \Phi\left(-1.64\right) \\ &= .95 \end{aligned}$$

Thus, we will say we are "confident" that θ_i is positive when $X_i \geq 2.32$. Thus, we can calculate the portion of the time we are confident that $\theta_i > 0$ but are actually incorrect as

$$\begin{aligned} P(\theta_i < 0 \,|\, X_i \geq 2.32) &= \frac{P(x_i \geq 2.32, \theta_i < 0)}{P(x_i \geq 2.32)} \\ &= \frac{\int_{-\infty}^{0} \Phi(\theta_i - 2.32)\phi(\theta_i)d\theta_i}{\int_{-\infty}^{\infty} \Phi(\theta_i - 2.32)\phi(\theta_i)d\theta_i} \\ &= .025 \end{aligned}$$

The last value is achieved by numerical integration. When I test this by simulation, drawing θ_i and then X_i for $n = 100,000$, I found that

$$\frac{|\{X_i > 2.32 \,\&\, \theta_i < 0\}|}{|\{X_i > 2.32\}|} = \frac{129}{5024} = .026$$

Confirming my result

iv) Part ii) confirms, as discussed in class, that a confidence interval can be viewed as the limit of a credible interval as the variance of a prior goes to infinity (assuming the support of the prior also covers the confidence interval).

3. i)

$$\begin{aligned} P(\pi \,|\, X_i, ..., X_n) &\propto P(X_i, ..., X_n \,|\, \pi)P(\pi) \\ &\propto \pi^{\sum_{i=1}^n X_i}(1-\pi)^{n-\sum_{i=1}^n X_i}\pi^{\alpha-1}(1-\pi)^{\beta-1} \\ &\propto \pi^{\alpha+\sum_{i=1}^n X_i - 1}(1-\pi)^{\beta+n-\sum_{i=1}^n X_i - 1} \\ &\sim \text{Beta}\left(\alpha + \sum_{i=1}^n X_i, \beta + n - \sum_{i=1}^n X_i\right) \end{aligned}$$

Thus, we can derive the distribution for the prediction of X_{n+1}

$$\begin{aligned} P(X_{n+1} = 1 \,|\, X_i, ..., X_n) &= \int_0^1 P(X_{n+1} = 1 \,|\, \pi, X_i, ..., X_n)P(\pi \,|\, X_i, ..., X_n)d\pi \\ &= \int_0^1 P(X_{n+1} = 1 \,|\, \pi)P(\pi \,|\, X_i, ..., X_n)d\pi \\ &= \int_0^1 \pi\pi^{\alpha+\sum_{i=1}^n X_i - 1}(1-\pi)^{\beta+n-\sum_{i=1}^n X_i - 1}d\pi \\ &= \frac{\alpha + \sum_{i=1}^n X_i}{\alpha + \sum_{i=1}^n X_i + \beta + n - \sum_{i=1}^n X_i} \\ &= \frac{\alpha + \sum_{i=1}^n X_i}{\alpha + \beta + n} \end{aligned}$$

So $X_{n+1} \,|\, X_i, ..., X_n \sim \text{Bernoulli}\left(\frac{\alpha+\sum_{i=1}^n X_i}{\alpha+\beta+n}\right)$.

ii)

$$\begin{aligned}
P(\pi \,|\, X_i, ..., X_n, w, \alpha, \beta) &\propto P(X_i, ..., X_n \,|\, \pi) P(\pi \,|\, w, \alpha, \beta) \\
&\propto \pi^{\sum_{i=1}^{n} X_i} (1-\pi)^{n-\sum_{i=1}^{n} X_i} \sum_{j=1}^{k} w_j \pi^{\alpha_j - 1} (1-\pi)^{\beta_j - 1} \\
&\propto \sum_{j=1}^{k} w_j \pi^{\alpha_j + \sum_{i=1}^{n} X_i - 1} (1-\pi)^{\beta_j + n - \sum_{i=1}^{n} X_i - 1} \\
&\sim \sum_{j=1}^{k} w_j \text{Beta}\left(\alpha_j + \sum_{i=1}^{n} X_i, \beta_j + n - \sum_{i=1}^{n} X_i - 1 \right)
\end{aligned}$$

Giving us a mixture of Betas for the posterior. The posterior mixing weights have remained the same, but the posterior hyperparameters have been updated as they would normally be for a Beta prior.

4. (a) I swapped code with Jonathan Eskreiswinkler.

(b/c) To start with, we seemed to have a large discrepancy - I had an accuracy around 77% and he had an accuracy around 84%. The difference turned out to be a few bugs on both our parts.

- I accidentally was dropping observations which were missing the allele at a particular locus
- For my likelihood, I was using P(Population | Allele) rather than P(Allele | Population)
- Jon was multiplying the population likelihoods by the wrong value for a few locus due to a bug in matching the appropriate values
- Jon accidentally multiplied by the likelihood for the first version of the locus twice, rather than the first and the second version (Ogo2 and Ogo2.1 for example).

In the end, both my version of the code and Jon's with my fixes had an accuracy of 78.0%.

(d) Besides the bugs, both Jon and I attempted to do the theoretical aspects identically. In terms of programing though, we had somewhat different approaches. I essentially had two arrays, one with all the probabilities from the training data set, and one with all the test data and associated priors, likelihoods, posteriors, and predictions. Jon generally stored these in separate arrays. Also, I tried to preform my calculations over the whole array at the same time to the extent possible, while Jon looped over each individual entry to do calculations. This made his code somewhat simpler, but mine is much more compact and runs quicker. I find my version a little easier to read, but then again, I wrote it. I didn't learn any R tricks from Jon.

Improved Code:

```
# Call given code
source('official/exercises/seeb/train_test.R')
subpops <- unique(test$Population)
loci <- names(test)[3:26]
loci <- locs[!grepl("\\.1",loci)] # get rid of .1 version

# Function to summarize frequency at locus by population (similar to trainc)
# Add 1 so that no Allele is impossible in each subpopulation
compute_freq <- function(data,locus){
    counts <- table(data[,locus],data$Population) +
                table(data[,paste(locus,".1",sep="")],data$Population) + 1
    return(counts/colSums(counts))
}

# Get frequency at each locus
train_freq <- list()
for (loc in loci) {
    train_freq[[loc]] <- as.data.frame.matrix(compute_freq(train,loc))
    train_freq[[loc]]$allele <- as.factor(rownames(train_freq[[loc]]))
}

# Set uniform prior
priors<-paste(subpops,"prior",sep="_")
test[,priors]<-.25

# Calculate Log-Likelihood
log_lks <-paste(subpops,"loglk",sep="_")
test[,log_lks]<-0
for (loc in loci) {
    for (e in c("",".1")) {
        locus <- paste(loc,e,sep="")
        test <- merge(test,train_freq[[loc]],by.x=locus,
                      by.y="allele",all.x=T,sort=F)
        # Missing data won't effect likelihood
        test[is.na(test[,locus]),subpops] <- 1
        test[,log_lks] <- test[,log_lks] + log(test[,subpops])
        test<-test[,!(names(test) %in% subpops)]

    }
}

# Calculate Posterior
posteriors<-paste(subpops,"post",sep="_")
test[,posteriors] <- test[,priors]*exp(test[,log_lks])
test[,posteriors] <- test[,posteriors]/apply(test[,posteriors],1,sum) # normalize

# Predict population with highest posterior
test$post_max <- apply(test[,posteriors],1,max)
test$predict <- ""
for (pop in subpops) {
    test[test$post_max==test[,paste(pop,"post",sep="_")],"predict"] <- pop
}

# Accuracy
accuracy <- mean(test$predict==test$Population)
```