Homework 1

Gaurav Kandlikar & Camila Madeiros

October 8, 2016

Question 1

1. Explain what your measurements will be.
   We will be exploring Gaurav's running speeds from his last ten runs.

Question 2

2. Deciding on some priors
   We estimate a mean speed (μ_0) of 8 minutes and 10 seconds (i.e. 490 seconds) per mile. The basis for this is that I generally try to run 8-minute (480 seconds) miles, and I know that over the past ~2 weeks I've been running slower due to a weird knee. We estimate a standard deviation (τ) of 30 seconds because I've done both road runs and treadmill runs, and I know that I run somewhat differently in the two conditions.

Question 3

3. Report the data and sample mean and variance (n-1) denominator.

```r
runs <- read.csv("running_data.csv")
head(runs)
```

```
##   sample distance time_min time_sec speed_sec_per_mi
## 1      1      3.5       27     1620              463
## 2      2      3.1       24     1440              465
## 3      3      7.1       58     3480              490
## 4      4      4.0       35     2100              525
## 5      5      5.4       48     2880              533
## 6      6     13.5      130     7800              578
```

```r
sample_mean <- mean(runs$speed_sec_per_mi)
sample_mean # get the sample mean
```

```
## [1] 514
```

```r
# Calculate sample variance
sample_var <- sum((runs$speed_sec_per_mi-sample_mean)^2)/(nrow(runs)-1)
```

The mean of our sample is 513.719; the standard deviation of our sample is 2133.818.

Question 4

4. Now specify the sampling standard deviation. Since we are doing a one parameter model, and since this value is usually not known, we need to do something because we are working with such a simple model.

We know that the speed estimates from the run tracking app are fairly accurate: the speeds from the app have closely matched my race speeds recorded independently. We don't think that the sampling standard deviation is higher than the sample σ of 46.193, so we will proceed in the analysis assuming the sampling standard deviation is the same as the sd of the data .

Question 5

5. Calculate the posterior mean, variance, and SD.

WORDS! Write the formulas for posterior mean etc.

```
prior_mean = 490 #seconds
prior_sd   = 30 #seconds
sampling_sd = sqrt(sample_var)
n = nrow(runs)
ybar = sample_mean

# calculate the posterior mean using the formula
posterior_mean <- (n/(sampling_sd^2))/((n/sampling_sd^2)+(1/prior_sd^2))*ybar +
  (1/prior_sd^2)/((n/sampling_sd^2)+(1/prior_sd^2)) * prior_mean

# calculate the posterior variance using the formula
posterior_var <- ((n/(sampling_sd^2))+(1/(prior_sd^2)))^-1
posterior_sd <- sqrt(posterior_var)
```

The posterior mean is $\bar{\mu} = 509.173$;
The posterior var is $V = 172.487$;
The posterior sd is $sd = 13.133$.

Question 6

6. The prior predictive density is the density that you predict for a single observation before seeing any data.
   It sure is!

```
prior_mean
```

```
## [1] 490
```

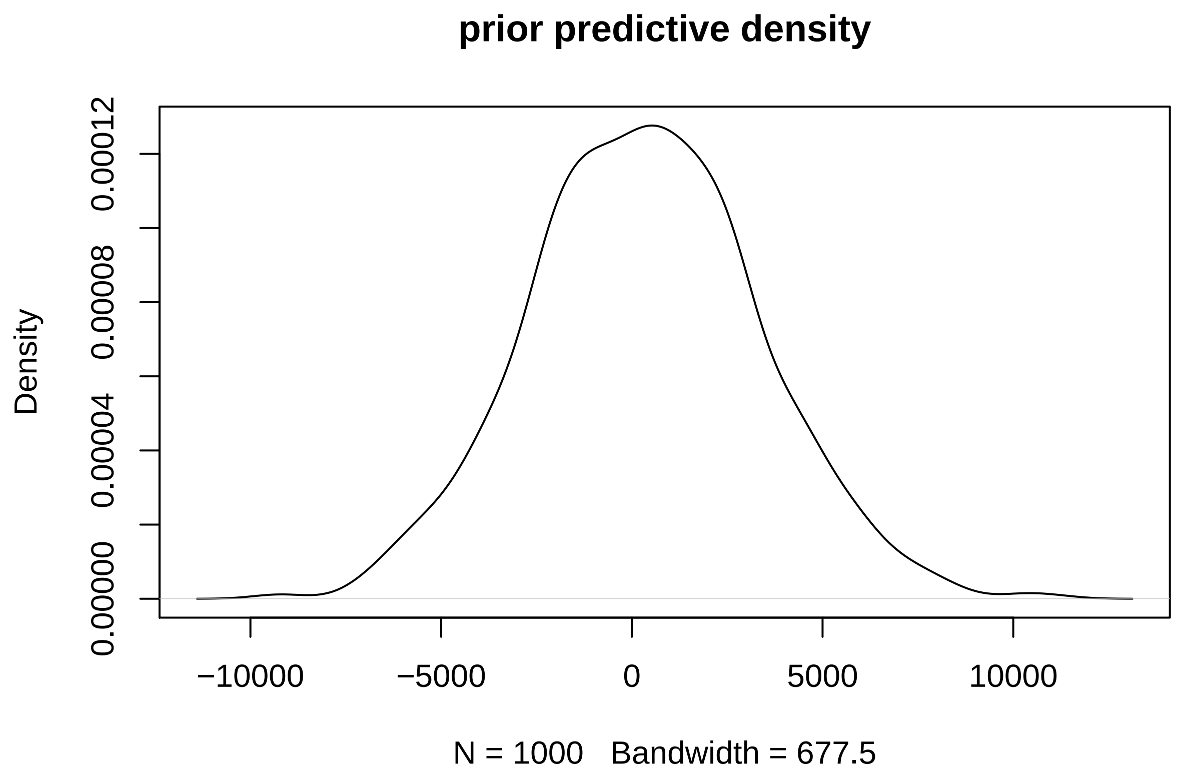
```
prior_sd
```

```
## [1] 30
```

```r
sampling_sd
```

```
## [1] 46.2
```

```r
plot(density(rnorm(1000, prior_mean, prior_sd^2+sampling_sd^2)),
     main = "prior predictive density")
```



Question 7

7. Construct a table with means, SDs and vars for the (i) posterior for mu, (ii) the prior for mu, (iii) the prior predictive for y, and (iv) the likelihood of mu.

```r
posterior_row  <- c(posterior_mean, posterior_sd, posterior_var)
prior_row      <- c(prior_mean, prior_sd, prior_sd^2)
prior_pred_row <- c(prior_mean, sqrt(prior_sd^2+sampling_sd^2),
                    prior_sd^2+sampling_sd^2)
likelihood_row <- c(sample_mean, sqrt(sample_var/n), sample_var/n)

to_print <- rbind(posterior_row, prior_row, prior_pred_row, likelihood_row)
rownames(to_print) <- c("Posterior", "Prior", "Prior Predictive", "Likelihood")
colnames(to_print) <- c("Mean", "SD", "Variance")
knitr::kable(to_print)
```

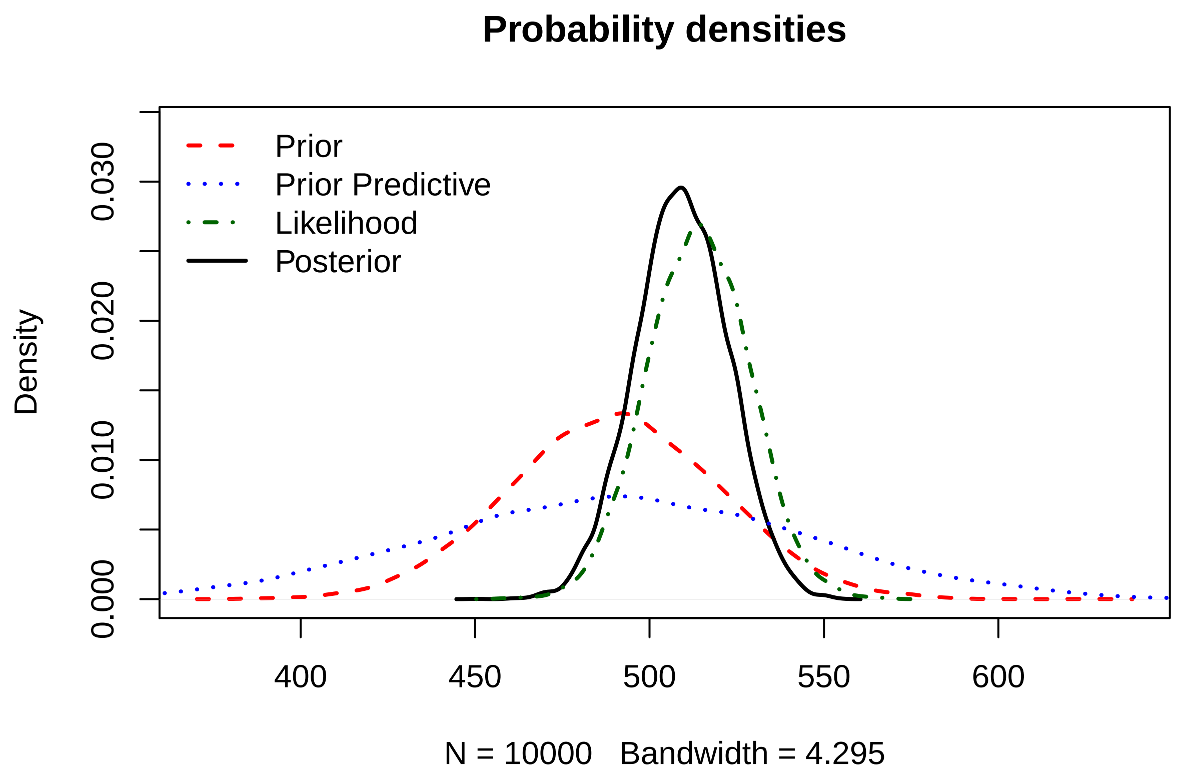
	Mean	SD	Variance
Posterior	509	13.1	172
Prior	490	30.0	900
Prior Predictive	490	55.1	3034
Likelihood	514	14.6	213

Question 8

8. Plot on a single plot the (i) posterior for mu, (ii) the prior for mu, (iii) the prior predictive for y, and (iv) the likelihood of mu (suitably normalized so it looks like a density, ie a normal with mean y-bar and variance sigma2 /n) all on the same graph. Interpret the plot.

```
posterior_vec <- rnorm(10000, posterior_mean, posterior_sd)
prior_vec <- rnorm(10000, prior_mean, prior_sd)
prior_pred_vec <- rnorm(10000, prior_mean, sqrt(prior_sd^2+sampling_sd^2))
likelihood_vec <- rnorm(10000, sample_mean, sqrt(sample_var/n))

plot(density(prior_vec), lty = 2, col = "red", main = "Probability densities",
     ylim = c(0, 0.034), lwd = 2)
lines(density(posterior_vec), lwd = 2)
lines(density(prior_pred_vec), col = "blue", lty = 3, lwd = 2)
lines(density(likelihood_vec), col = "darkgreen", lty = 4, lwd = 2)
legend("topleft", lty = c(2, 3, 4, 1), col = c("red", "blue", "darkgreen", "black"), lwd = 2,
       legend = c("Prior", "Prior Predictive", "Likelihood", "Posterior"), bty = "n")
```



Interpretation of plot: The posterior distribution of mean running speed is in between the prior mean and the likelihood estimate; in other words, the posterior is a compromise between the likelihood and the prior, which shrank the likelihood. The prior predictive has a mean equal to the prior mean but has a much wider distribution, which comes from our uncertainty in measurements as well as the prior estimate for variation in running speed. Our posterior distribution is quite similar to the likelihood distribution, which happens because the weight given to our prior mean is lower than the weight given to the sample mean in the posterior calculation.

Question 9

9. 9. Write R/WinBUGS programs to sample from the posterior of mu.

```
# sink("running_model.txt")
cat("model {
    for (i in 1:N) {
        x[i] ~ dnorm(mu, tau)
    }
    mu ~ dnorm(prior_mean, prior_tau) # change this to dnorm(prior_mean and prior_tau)
    sigma <- sampling_sd              # change this to be sampling_sd
    tau <- pow(sigma, -2)             # tau equal to 1/sigma^2
}", fill = TRUE)
```

```
## model {
##  for (i in 1:N) {
##      x[i] ~ dnorm(mu, tau)
##  }
##  mu ~ dnorm(prior_mean, prior_tau) # change this to dnorm(prior_mean and prior_tau)
##  sigma <- sampling_sd              # change this to be sampling_sd
##  tau <- pow(sigma, -2)             # tau equal to 1/sigma^2
## }
```

```
# sink()
```

```
# parameters
jags.params = c("mu", "sigma", "tau")

# data
x = runs$speed_sec_per_mi
N = length(runs$speed_sec_per_mi)
prior_tau = 1/(prior_sd^2)
jags.data = list("x", "N", "prior_mean", "sampling_sd", "prior_tau")

# initials

jags.inits = function(){
    list("mu" = 0) # part of algorithm!
}
```

Now that we have set up the model, data, and parameters, we can run the model:

```
hw1.sim = jags(jags.data, jags.inits, jags.params,
              model.file = "running_model.txt",
              n.chains = 3, n.iter = 11000, n.burnin = 1000)
```

```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 10
##    Unobserved stochastic nodes: 1
##    Total graph size: 19
##
## Initializing model
```

We can summarize the output of the model:

```
to_print <- hw1.sim$BUGSoutput$summary[2:4,c("mean", "sd", "2.5%", "97.5%")]
knitr::kable(to_print)
```

	mean	sd	2.5%	97.5%
mu	509.4	12.8	484.7	534.4
sigma	46.2	0.0	46.2	46.2
tau	0.0	0.0	0.0	0.0

Question 10

10. Adapt your BUGS program to sample from the prior and prior predictive. Do this by not loading your data, rather, in loading the initial values, move the data y over to the init list instead. There is an example at the end of Homework 2 for a Poisson-gamma likelihood/prior. [Helpful step: set keyword DIC=F in the call to bugs, as WinBUGS can not calculate DIC for prior predictions.]

```
# sink("running_model2.txt")
cat("model {
    for (i in 1:N) {
        x[i] ~ dnorm(mu, tau)
    x2[i] ~ dnorm(mu, tau2)
    }
    mu  ~ dnorm(prior_mean, prior_tau) # change this to dnorm(prior_mean and prior_tau)
    mu2 ~ dnorm(prior_mean, prior_tau) # prior predictive
    sigma <- sampling_sd # change this to be sampling_sd
    tau <- pow(sigma, -2) # tau equal to 1/sigma^2
  tau2 <- tau + prior_tau
}", fill = TRUE)
```

```
## model {
##   for (i in 1:N) {
##       x[i] ~ dnorm(mu, tau)
##     x2[i] ~ dnorm(mu, tau2)
##   }
##   mu  ~ dnorm(prior_mean, prior_tau) # change this to dnorm(prior_mean and prior_tau)
##   mu2 ~ dnorm(prior_mean, prior_tau) # prior predictive
##   sigma <- sampling_sd # change this to be sampling_sd
##   tau <- pow(sigma, -2) # tau equal to 1/sigma^2
##    tau2 <- tau + prior_tau
## }
```

```
# sink()
```

```
jags.data2 = list("N", "prior_mean", "sampling_sd", "prior_tau")

# initials
jags.params = c("mu", "sigma", "tau")

jags.inits2 = function(){
    list("x" = x, "mu" = 0) # part of algorithm!
}

hw1.sim2 = jags(jags.data2, jags.inits2, jags.params,
              model.file = "running_model2.txt",
              n.chains = 3, n.iter = 11000, n.burnin = 1000, DIC = F)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
```

```
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 22
##    Total graph size: 31
##
## Initializing model
```