JC STAT 638 HW 9

9.1)

```r
swim <- read.table("swim.dat")
head(swim)
```

```
    V1   V2   V3   V4   V5   V6
1 23.1 23.2 22.9 22.9 22.8 22.7
2 23.2 23.1 23.4 23.5 23.5 23.4
3 22.7 22.6 22.8 22.8 22.9 22.8
4 23.7 23.6 23.7 23.5 23.5 23.4
```

Pivoting the data:

```r
library(tidyverse)
colnames(swim) <- c(2,4,6,8,10,12)
swim_long <- swim |>
  mutate(Swimmer = row_number()) |>
  pivot_longer(
    cols = -Swimmer,
    names_to = "Week",
    values_to = "Time"
  ) |>
  mutate(Week = as.integer(Week))
```

```r
head(swim_long)
```

```
# A tibble: 6 x 3
  Swimmer  Week  Time
```

```
    <int> <int> <dbl>
1       1     2  23.1
2       1     4  23.2
3       1     6  22.9
4       1     8  22.9
5       1    10  22.8
6       1    12  22.7
```

a)

Since competitive times for this age group range from 22 to 24 seconds we choose $N(23, 1)$ as the prior for the intercept term β_0.

There are two weeks between each swim so there wouldn't be any effect of fatigue on the swimmers. So for a selection β_1 I chose $N(0, .1)$ because we wouldn't expect much change over a 12 week window.

I also assume that β_0 and β_1 are independent.

For variance I selected a weak prior as we don't have information about how much error to expect. So for this I used the unit information prior, taking ν_0 and $\sigma_0^2 = \hat{\sigma}^2_{ols}$, where $\hat{\sigma}^2_{ols}$ I get from fitting the linear model $\text{Time} = \beta_0 + \text{Week} + \text{Swimmer} + e$.

```r
lin_fit <- lm(Time ~ Week + Swimmer, data = swim_long)
summary(lin_fit)
```

```
Call:
lm(formula = Time ~ Week + Swimmer, data = swim_long)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.6361 -0.2484  0.0800  0.2508  0.4277 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 22.862500   0.218149 104.802   <2e-16 ***
Week        -0.005357   0.019998  -0.268   0.7914    
Swimmer      0.131667   0.061094   2.155   0.0429 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3346 on 21 degrees of freedom
```

```
Multiple R-squared:  0.1834,    Adjusted R-squared:  0.1056
F-statistic: 2.358 on 2 and 21 DF,  p-value: 0.1191
```

So our selection of $\sigma_0^2 = .3346^2$.

```
sigma2_0 <- sum(resid(lin_fit)^2)/(24 - 3);sigma2_0
```

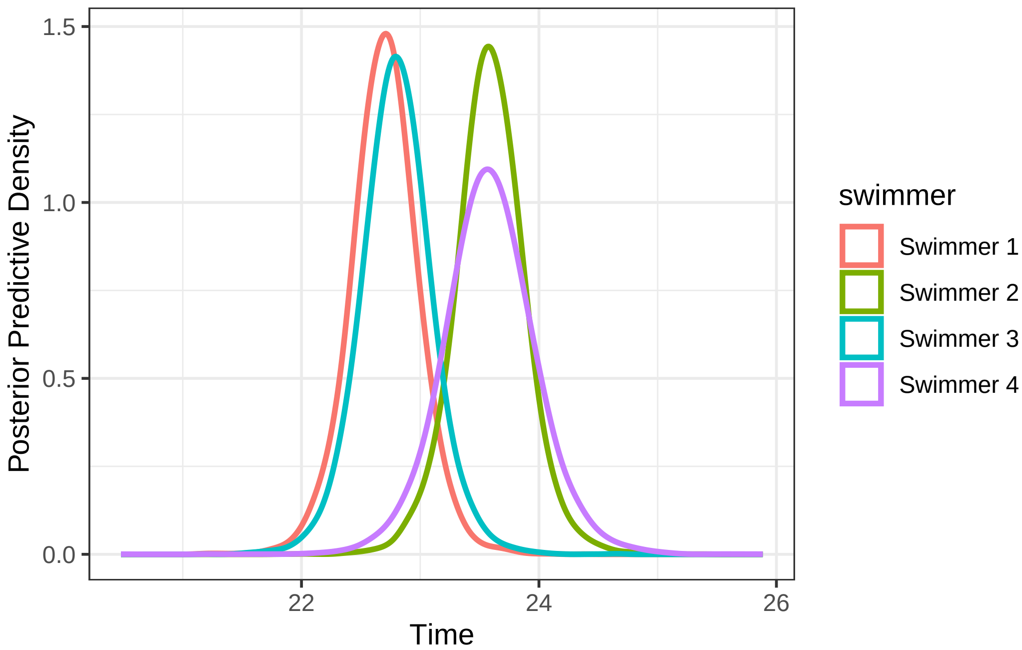
```
[1] 0.1119745
```

Getting posterior predictive distributions:

```
set.seed(10)
library(MASS)
#Priors
m0 <- c(23, 0)
V0 <- diag(c(1,.1))
V0inv <- solve(V0)
nu_0 <- 1

predict_two_weeks <- function(df) {
  y  <- df$Time
  X  <- cbind(1, df$Week)
  n  <- length(y)

  # Posterior parameters
  Vn_inv <- V0inv + t(X)%*%X
  Vn     <- solve(Vn_inv)
  mn     <- Vn %*% (V0inv %*% m0 + t(X)%*%y)

  an <- nu_0 + n/2
  bn <- sigma2_0 + 0.5*( t(m0)%*%V0inv%*%m0 + t(y)%*%y - t(mn)%*%Vn_inv%*%mn )

  # Prediction
  w_new <- max(df$Week) + 2
  x_new <- c(1, w_new)

  # Sampling
  S <- 10000
  ypred <- numeric(S)

  for (s in 1:S) {
```

```r
    sigma2 <- 1 / rgamma(1, an, bn)  # Inv-gamma
    beta   <- mvrnorm(1, mn, sigma2*Vn)
    ypred[s] <- rnorm(1, x_new %*% beta, sqrt(sigma2))
  }

  return(ypred)
}

pred_post_dist <- lapply(split(swim_long, swim_long$Swimmer), predict_two_weeks)
```

```r
library(tidyverse)
pred_df <- map2_df(pred_post_dist, 1:length(pred_post_dist),
                          ~ tibble(
                            swimmer = paste0("Swimmer ", .y),
                            ypred   = .x
                          ))
pred_df |>
  ggplot(aes( x = ypred, color = swimmer)) +
  geom_density(adjust = 2, linewidth = 1) +
  labs(
    x = "Time",
    y = "Posterior Predictive Density"
  ) +
  theme_bw()
```



b)

We find the swimmer with the longest time at each draw.

```
S <- length(pred_post_dist[[1]])
pred_mat <- do.call(cbind, pred_post_dist)
slowest <- apply(pred_mat, 1, which.max)
prob_slowest <- table(slowest)/S
prob_slowest
```

```
slowest
     1      2      3      4
0.0039 0.4987 0.0083 0.4891
```

If we were concerned only with not sending the slowest swimmer we would send the first swimmer. I think its worth looking at who the fastest runner is at each draw to see if that gives us the same result.

```
fastest <- apply(pred_mat,1,which.min)
prob_fastest <- table(fastest)/S
prob_fastest
```

```
fastest
     1      2      3      4
0.5858 0.0053 0.3926 0.0163
```

The first swimmer is most likely to have the fastest swim as well, I would recommend sending this swimmer to the competition.