11.3

Given the following data:

```
Data                 <-  data.frame(machine = rep(1:6, each=5),
                                     measure = c(83, 92, 92, 46, 67,
                                                  117, 109, 114, 104, 87,
                                                  101, 93, 92, 86, 67,
                                                  105, 119, 116, 102, 116,
                                                  79, 97, 103, 79, 92,
                                                  57, 92, 104, 77, 100))
J                 <- length(unique(Data$machine))
n                 <- length(Data$measure)
```

Starting points
Select 10 θ_j randomly from the y_{ij} sample.

```
theta_start          <- sapply(1:6,function(x) sample(Data$measure[Data$machine==x],
                                                      10, replace=TRUE))
```

1. Conditional posterior of τ^2

```
TauHat2       <- function(theta) {
  mu              <- mean(theta)
  Tau2          <- ( 1/(J-1) ) * sum((theta - mu)^2)
  return(Tau2)
}

Tau_CondPosterior      <- function(theta) {
  Tau2          <- TauHat2(theta)
  Tau_Post       <- (J - 1) * (Tau2)/rchisq(1,J-1)
  return(Tau_Post)
  }
```

2. Conditional posterior of σ^2

```
FsigmaHat2   <- function(theta) {
  sigmaHat2   <-  sapply(1:6, function(x) (Data$measure[Data$machine==x] - theta[x])^2)
  sigmaHat2   <-  (1/n) * sum(unlist(sigmaHat2))
  return(sigmaHat2)
}

Sigma2_CodPosterior      <-  function(theta) {
  Sigma2_Hat      <- FsigmaHat2(theta)
  Sigma2_Post      <-  (n) * (Sigma2_Hat)/rchisq(1,n)
  return(Sigma2_Post)
  }
```

3. Conditional posterior of μ

```
Mu_Hat          <- function(theta) {
  mean(theta)
}

Mu_CondPosterior        <- function(theta,tau2) {
  Mu_Hat    <- Mu_Hat(theta)
  rnorm(1,Mu_Hat,sqrt(tau2/J))
}
```

4. Conditional posterior of θ

```
Theta_Hat_j     <- function(j,mu,sigma2,tau2) {
  Yavg_j       <- mean(Data$measure[Data$machine==j])
  N_j           <- length(Data$measure[Data$machine==j])
  ( (1/tau2) * mu + (N_j/sigma2) * Yavg_j  ) / ( (1/tau2) + (N_j/sigma2)  )
}
VTheta_Hat_j   <- function(j,mu,sigma2,tau2) {
  N_j           <- length(Data$measure[Data$machine==j])
  ( 1 ) / ( (1/tau2) + (N_j/sigma2) )
}
Theta_CondPosterior         <- function(mu,sigma2,tau2) {
  theta         <- NULL
  for (j in 1:J) {
    T_Hat       <- Theta_Hat_j(j,mu,sigma2,tau2)
    V_T_Hat     <- VTheta_Hat_j(j,mu,sigma2,tau2)
    theta[j]    <- rnorm(1,T_Hat,sqrt(V_T_Hat))
  }
  return(theta)
}
```

** Hierarchical Model: Sample using Gibbs Sampler **

```r
sims          <- 200
Gibbs_Sampler      <- function(start_pt_seq_No) {
  param         <- 9
  sample_parameters       <- matrix(NA, ncol = param, nrow = sims )
  colnames(sample_parameters)<-  c("theta1", "theta2", "theta3",
                          "theta4", "theta5", "theta6",
                          "mu", "sigma2", "tau2")
  sample_parameters[1,1:6]<- theta_start[start_pt_seq_No,]
  sample_parameters[1,9]  <- Tau_CondPosterior(theta_start[start_pt_seq_No,])
  sample_parameters[1,8]  <- Sigma2_CodPosterior(theta_start[start_pt_seq_No,])
  sample_parameters[1,7]  <- Mu_CondPosterior(theta_start[start_pt_seq_No,],sample_param
eters[1,9])

  for (s in 2:sims) {
    sample_parameters[s,1:6]<- Theta_CondPosterior(sample_parameters[s-1,7],sample_param
eters[s-1,8],sample_parameters[s-1,9])
    sample_parameters[s,9]  <- Tau_CondPosterior(sample_parameters[s,1:6])
    sample_parameters[s,8]  <- Sigma2_CodPosterior(sample_parameters[s,1:6])
    sample_parameters[s,7]  <- Mu_CondPosterior(sample_parameters[s,1:6],sample_paramete
rs[s,9])
  }
return(sample_parameters)
#Warm-up
}
sample_parameters           <- lapply(1:10, function(x) Gibbs_Sampler(x))
sample_parameters.1         <- sample_parameters[[1]][101:200, ]
sample_parameters.combined <- rbind(sample_parameters[[1]][101:200, ], sample_parameters
[[2]][101:200, ], sample_parameters[[3]][101:200, ], sample_parameters[[4]][101:200, ],
 sample_parameters[[5]][101:200, ], sample_parameters[[6]][101:200, ], sample_parameters
[[7]][101:200, ], sample_parameters[[8]][101:200, ], sample_parameters[[9]][101:200, ],
 sample_parameters[[10]][101:200, ])
#Transform the variance in to sd.
sample_parameters.combined[,8:9]   <- sqrt(sample_parameters.combined[,8:9] )

# Quantiles
t(apply(sample_parameters.combined,2, function(x) quantile(x, c(.025,.25,.5,.75,.975))))
```

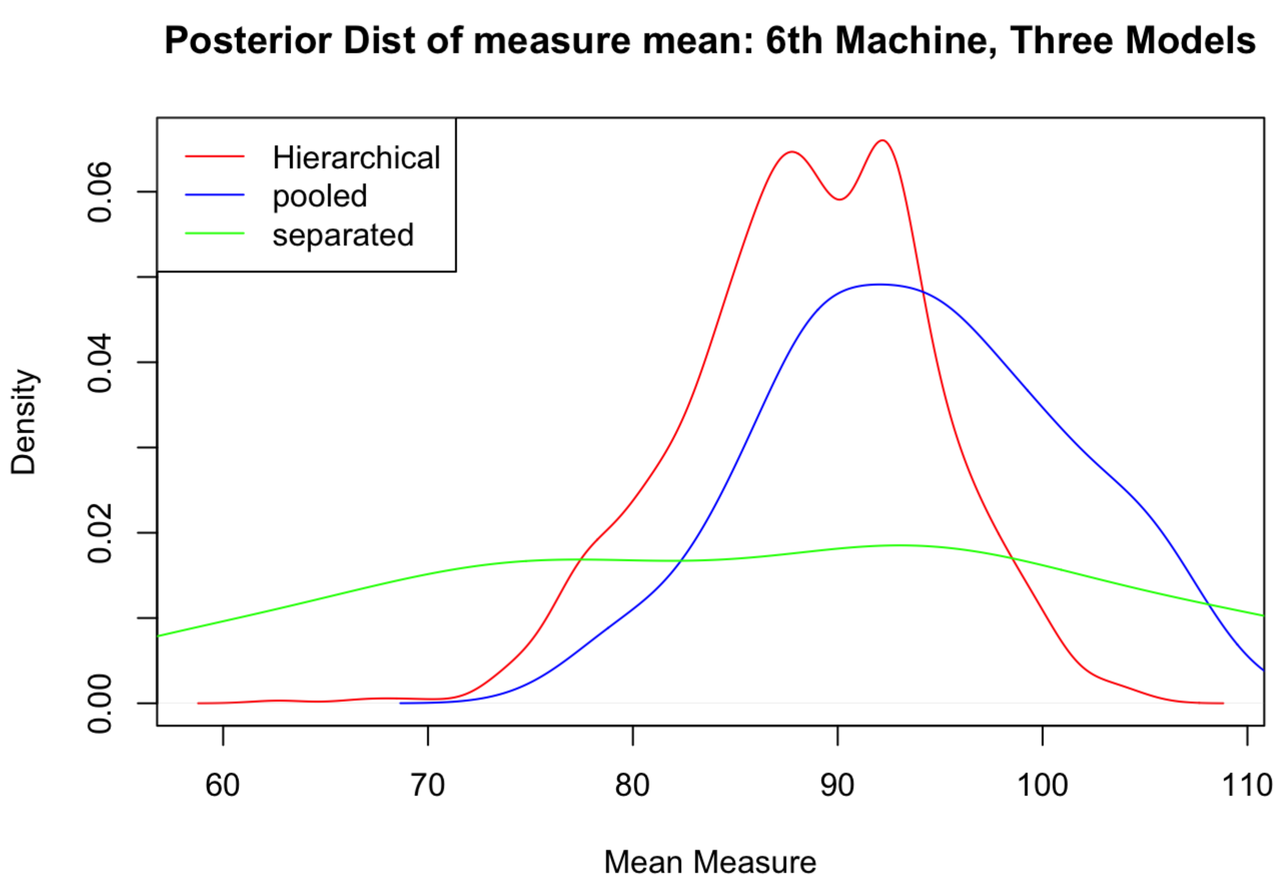
```
##               2.5%       25%       50%       75%     97.5%
## theta1 69.33451180 77.057104  81.58411  86.78668  95.51058
## theta2 89.45782851 97.007670 101.53710 106.49968 114.83650
## theta3 77.27804552 85.883826  90.07689  92.98418 100.38957
## theta4 91.84875536 99.706566 105.39575 110.58712 119.11883
## theta5 79.18944654 87.732601  91.87743  95.24059 102.29602
## theta6 76.44598989 84.747576  88.72221  92.49478  99.68702
## mu     80.35995763 89.911681  92.77493  96.96565 107.01125
## sigma2 11.23766052 13.459499  14.97195  16.77308  21.32504
## tau2    0.03335087  7.141435  10.65701  14.98413  30.76660
```

** Pooled and Separated Model **

```
Yavg_Pooled   <-  mean(Data$measure)
Var_Pooled      <-  sum( (Data$measure[Data$machine==6] -
                            mean(Data$measure))^2 )/( length(Data$measure) - 1)
Theta_Posterior_Pooled_6<- rnorm(sims, Yavg_Pooled, sqrt(Var_Pooled))
Yavg_Pooled_6     <-  mean(Data$measure[Data$machine==6])
Var_Pooled_6       <-  sum( (Data$measure[Data$machine==6] -
                            mean(Data$measure[Data$machine==6]))^2 ) /
                        ( length(Data$measure[Data$machine==6]) - 1)
Theta_Posterior_Sep_6   <- rnorm(sims, Yavg_Pooled_6, sqrt(Var_Pooled_6))
```

** Posterior distribution of the mean of the quality measurements of the sixth machine. For the hierarchical, pooled and separate models **

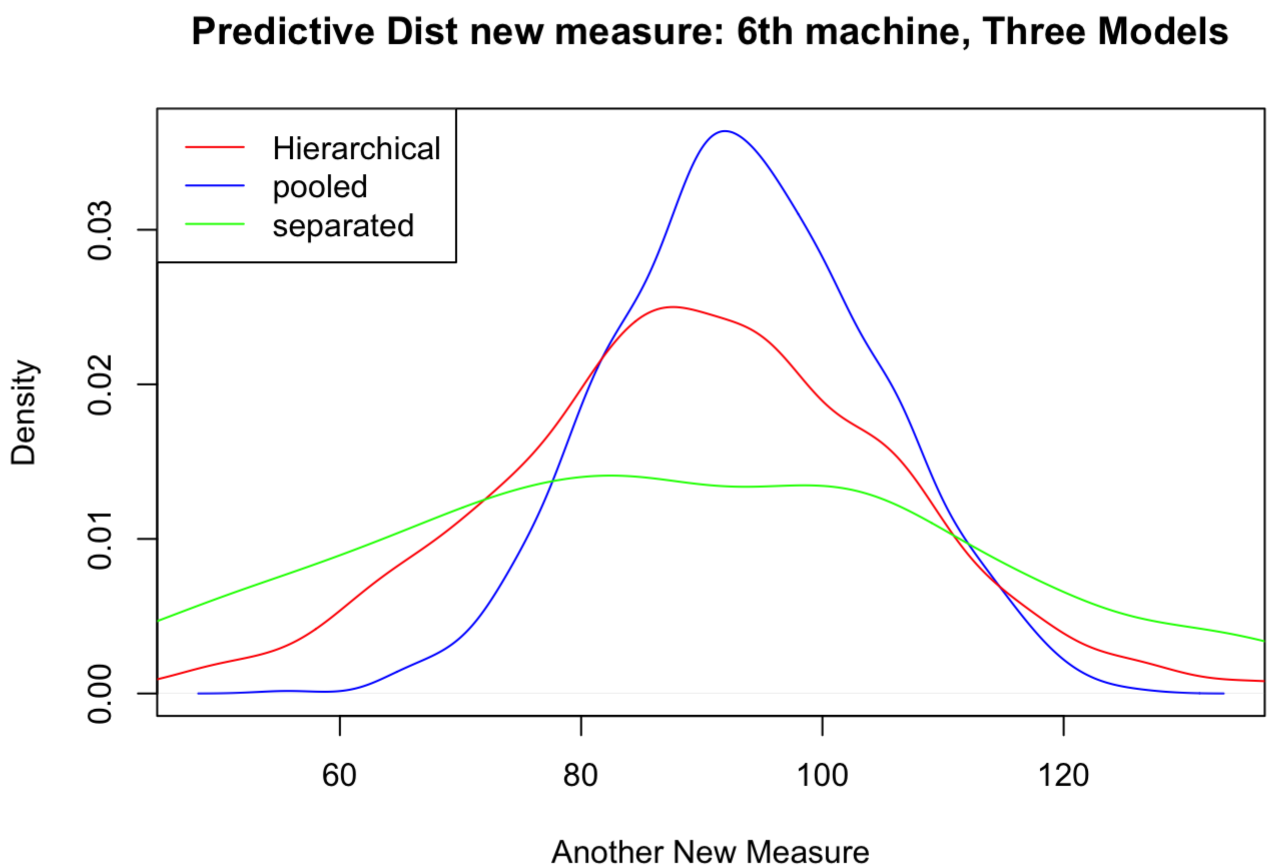
```
plot(density(sample_parameters.combined[,"theta6"]), col="red",
     xlab="Mean Measure",
     ylab="Density",
     main="Posterior Dist of measure mean: 6th Machine, Three Models")
lines(density(Theta_Posterior_Pooled_6), col="blue")
lines(density(Theta_Posterior_Sep_6), col="green")
legend("topleft",col = c("red","blue", "green"),
       legend=c("Hierarchical","pooled", "separated"),
       lty = c(1,1,1))
```



** Predictive distribution for another quality measurement of the sixth machine. **

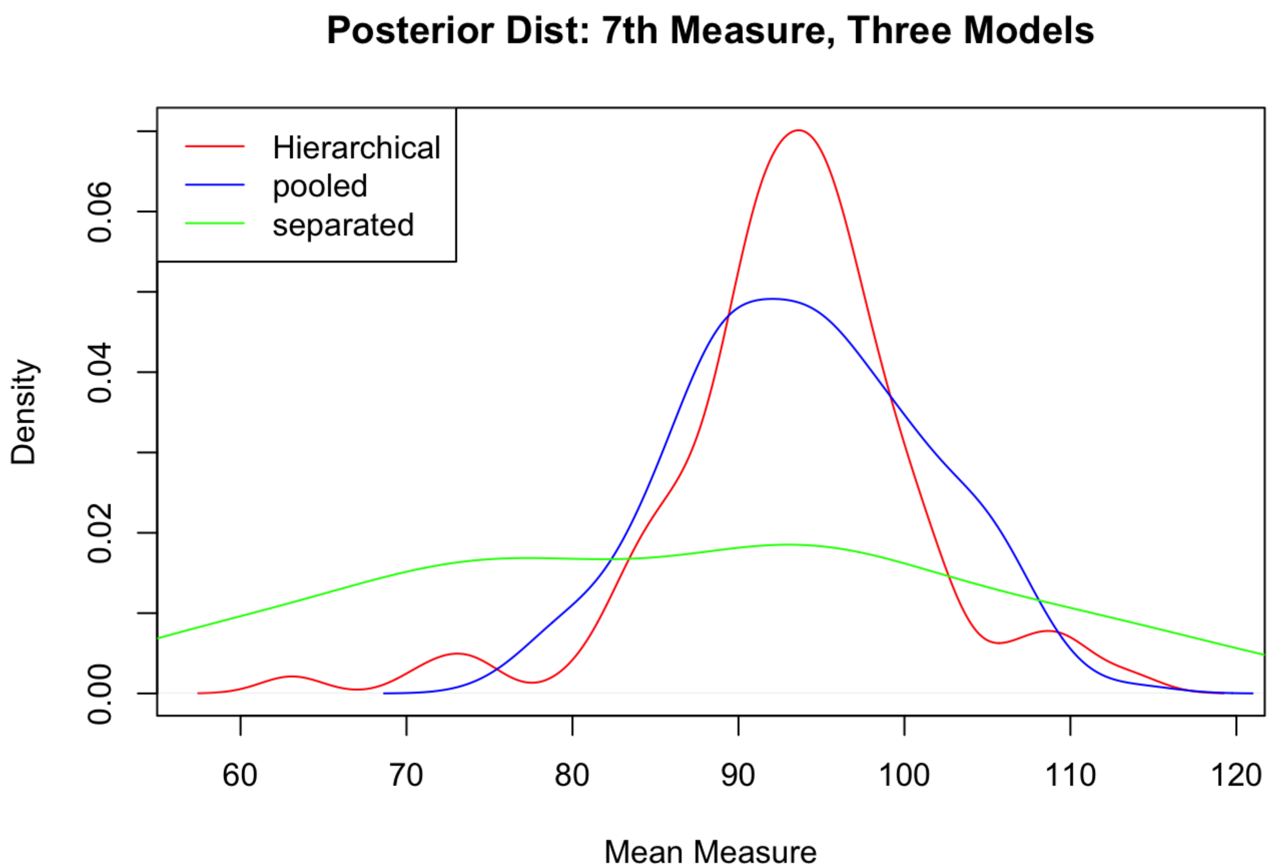
```
y_pred_Hierar <- rnorm(1000, mean=sample_parameters.combined[,"theta6"], sd = sample_par
ameters.combined[,"sigma2"])
y_pred_Pooled <- rnorm(1000, mean = Theta_Posterior_Pooled_6, sd = sqrt(Var_Pooled))
y_pred_Sep <- rnorm(1000, mean = Theta_Posterior_Sep_6, sd = sqrt(Var_Pooled_6))

plot(density(y_pred_Pooled), col= "blue",
     xlab="Another New Measure",
     ylab="Density",
     main="Predictive Dist new measure: 6th machine, Three Models")
lines(density(y_pred_Hierar), col="red")
lines(density(y_pred_Sep), col="green")
legend("topleft",col = c("red","blue", "green"),
       legend=c("Hierarchical","pooled", "separated"),
       lty = c(1,1,1))
```



** Posterior distribution of the mean of the quality measurements of the seventh machine. For the hierarchical, pooled and separate models **

```
plot(density(sample_parameters.1[,"mu"]), col="red", xlab="Mean Measure",
    ylab="Density", main="Posterior Dist: 7th Measure, Three Models")
lines(density(Theta_Posterior_Pooled_6), col="blue")
lines(density(Theta_Posterior_Sep_6), col="green")
legend("topleft",col = c("red","blue", "green"),
       legend=c("Hierarchical","pooled", "separated"),
       lty = c(1,1,1))
```


Posterior Dist: 7th Measure, Three Models
Hierarchical
pooled
separated
Density
0.00
0.02
0.04
0.06
60
70
80
90
100
110
120
Mean Measure