Supporting Materials for Lecture 10

We will revisit the example from lecture 4, where he had three variables: age, sex and systolic blood pressure. Whereas, sex and age are completely observed, the systolic blood pressure (SBP) variable has some missing values. I've eliminated the columns subject id and R, since they would be interpreted as variables when passed to function mice().

```
rm(list=ls())
require(mice)
datasbp=data.frame("Sex"=c(rep("Male",12),rep("Female",8)),
                   "Age"=c(50,41,52,58,56,45,42,48,57,56,55,58,57,44,56,45,48,50,59,55),
                   "SBP"=c(163.5,126.4,150.7,190.4,172.2,NA,136.3,146.8,162.5,161,148.7,
                          163.6,NA,140.6,NA,118.7,NA,104.6,131.5,126.9))
attach(datasbp)
```

Although not of great interest, the package `mice` has also a function to the mean imputation.

```
#mean imputation
step1=mice(datasbp,method="mean",m=1,seed=1)

##
##  iter imp variable
##   1   1  SBP
##   2   1  SBP
##   3   1  SBP
##   4   1  SBP
##   5   1  SBP

step1$imp$SBP

##           1
## 6   146.525
## 13  146.525
## 15  146.525
## 17  146.525
```

This is, of course, what we had obtained back in lecture 4. In this case does not make sense to specify m, the number of imputations, greater than 1, since they would all be equal (there is nothing stochastic here). Obviously, $m = 1$ corresponds to single imputation. Just for the sake of illustration, I will use $m = 2$.

```
step1=mice(datasbp,method="mean",m=2,seed=1)

##
##  iter imp variable
##   1   1  SBP
```

```
##    1    2  SBP
##    2    1  SBP
##    2    2  SBP
##    3    1  SBP
##    3    2  SBP
##    4    1  SBP
##    4    2  SBP
##    5    1  SBP
##    5    2  SBP


step1$imp$SBP


##          1        2
## 6   146.525 146.525
## 13  146.525 146.525
## 15  146.525 146.525
## 17  146.525 146.525
```

The same can be said about regression imputation, which can be implemented by letting the method argument to be equal to `norm.predict`.

```
step1=mice(datasbp,method="norm.predict",m=1,print=FALSE,seed=1)
step1$imp$SBP


##          1
## 6   144.8417
## 13  135.0316
## 15  133.3798
## 17  120.1653
```

The above obtained values are exactly the same we had obtained in slide 6 of lecture 4.
Suppose now that we will use stochastic regression imputation in the first step and that our model of interest (step 2), i.e., the model we would have used in case we had no missing data, is a linear model where the response is the SBP and the covariates are sex and age. Note that, in this (toy) example, the model for imputation and the model of interest coincide, however, this does not need to be the case.

```
step1=mice(datasbp,method="norm.nob",m=10,print=FALSE,seed=1)
#summary(step1)
step2=with(step1,lm(SBP~Sex+Age))
#summary(step2)
step3=pool(step2)
summary(step3)[,c(1,2,3,5)]


##                estimate   std.error statistic    p.value
## (Intercept) 36.041112 30.0189797  1.200611 0.25308776
## SexMale     28.430317  7.0706180  4.020910 0.00169982
## Age          1.762263  0.5873797  3.000212 0.01107272
```

After pooling the results of the 10 datasets together, we obtain the above estimates and standard errors for the regression coefficients. We can also compare the observed data (for SBP) and the imputed SBP values. The plot below compares the density of theobserved SBP values (in blue) with the ones of the imputed SBP values, for each of the 10 datasets. We expect them to be similar (though not identical) under MAR assumption.

```
densityplot(step1)
```
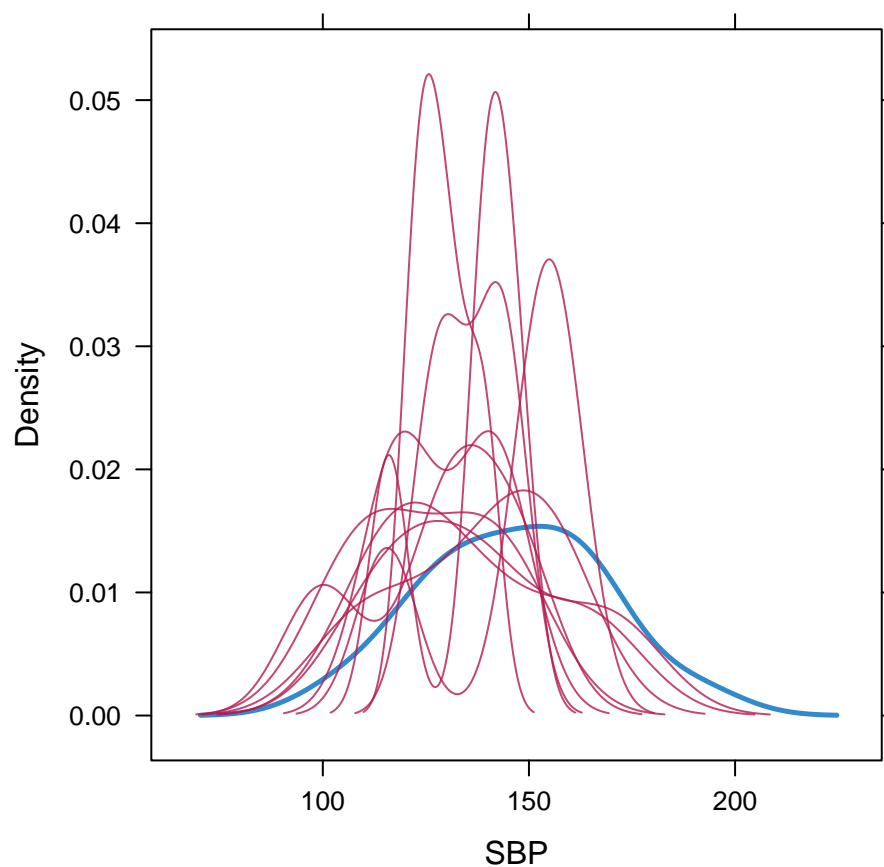

We will repeat the process but now using a Bayesian linear model in step 1.

```
step1=mice(datasbp,method="norm",m=10,seed=1,print=FALSE)
step2=with(step1,lm(SBP~Sex+Age))
step3=pool(step2)
summary(step3)[,c(1,2,3,5)]

##               estimate  std.error statistic     p.value
## (Intercept) 39.478408 32.2479326  1.224215 0.244273720
## SexMale     31.434838  8.8498519  3.552019 0.003959057
## Age          1.660962  0.6359673  2.611709 0.022668507
```

We will now consider the bootstrap alternative to the Bayesian normal linear model.

```
step1=mice(datasbp,method="norm.boot",m=10,seed=1,print=FALSE)
step2=with(step1,lm(SBP~Sex+Age))
step3=pool(step2)
summary(step3)[,c(1,2,3,5)]
```

```
##             estimate  std.error statistic    p.value
## (Intercept) 41.356785 29.4262371  1.405439 0.184798196
## SexMale     30.043446  8.1679303  3.678220 0.003068047
## Age          1.626932  0.5883617  2.765190 0.016864028
```