There is an R code set at the end of this lab and posted on the course web page that will prove useful for doing this lab. **Use it in answering/doing the following.** This lab concerns computing and factorial data sets.

**The First Part of the Lab Concerns More About 2-Way Analyses**

1. Make a data frame for the $3 \times 3$ factorial data in Table 8.1 of Vardeman and Jobe. Then set the "effects sum to 0" option for your R session.

2. Compute the $r = 9$ sample means and standard deviations for this data set. Be sure you understand the meaning of the syntax supplied in the Lab code set.

3. Make a one-way analysis of these data using `lm()` . How are the estimated "coefficients" and the "predictions" produced related to the sample means? Verify that the "Residual standard error" is our $s_r$ for this data set. There are values for "se.fit" available. What are these? (Give a formula.) Why are the se.fit values not all the same?

4. Make an interaction plot enhanced with error bars derived from 95% confidence limits for the $r = 9$ cell means. This should look much like Figure 8.3 of Vardeman and Jobe.

5. Make a two-way analysis of these data using `lm()` and a model including main effects of both wood and joint types and their interactions. Verify that the estimated "coefficients" are exactly as on calculated "by hand" on pages 166-170 of Vardeman and Jobe. Verify that the printed standard errors for the interactions are as suggested on pages 560-561 of Vardeman and Jobe. What are the "predictions" produced by this analysis?

6. What is a $p$-value for testing $H_0 : \alpha \beta_{ij} = 0 \ \forall i, j$?

7. Change the order in which the factors wood type and joint type are entered in the `lm()` call. What then changes in an ANOVA for these data?

8. These data are not balanced. As a result, estimates of effects and fitted values for the "main effects only" model are not simply related to the estimates of effects for the full model (where each cell has an unrestricted mean and the effects are as defined in the book and in class). Compare the fitted main effects from the full model to those produced by an `lm()` involving only wood type and joint type. Verify that the predictions for the latter are sums of coefficients from the latter, but are not sums of coefficients from the former.

9. Redo the two-way `lm()` call for the full model using numerical dummy variables instead of the "factors." Verify that coefficients and standard errors and ANOVAs all match as one expects.

**The Second Part of the Lab Concerns $2^p$ Analyses**
10. Make a data frame to hold the data of Table 8.8, page 569 of Vardeman and Jobe. Make Tool, Angle, and Cut "factors."

11. Use \texttt{lm()} to fit all $2^3$ factorial effects in this situation. How do the estimated coefficients compare to what is produced by the Yates algorithm? (See page 576 of Vardeman and Jobe.) How do the standard errors for the fitted coefficients compare to the plus or minus parts of displays (8.12) and (8.13) on page 575 of Vardeman and Jobe.

12. The data in this example are balanced (the $r = 8$ sample sizes are all the same). Because of this, fits made with reduced models produce estimated coefficients that agree with estimates from the fit made with the full model. Fit the "Angle and Cut main effects only" model to these data and verify that estimated coefficients agree with ones from the fit of the full model. Verify that the (reduced model) predicted values are therefore equal to sums of estimated coefficients from the full model.

13. Redo the two-way \texttt{lm()} call for the full model using numerical dummy variables instead of the "factors." Verify that coefficients and standard errors and ANOVAs all match as one expects.

\textbf{Code Set for Stat 401B Laboratory #10}

\texttt{
#Here is code for Stat 401B Lab 10
#Here is Some Code for Analyses of Factorial Data
#First a two-way example, Example 1 of Chapter 8 of Vardeman and Jobe
#Begin by creating a data frame

Joint<-c(2,1,2,1,2,2,3,2,1,3,3,1,3,1,3,1,2)
Wood<-c(2,1,3,2,2,1,3,3,2,2,1,1,1,3,1)
CaseType<-c(5,1,6,2,5,4,9,6,3,8,8,7,1,7,3,4)
FailStress<-
c(1518,829,2571,1169,1927,1348,1489,2443,1263,1295,1561,1000,
   596,859,1029,1207)
Joint<-as.factor(Joint)
Wood<-as.factor(Wood)
CaseType<-as.factor(CaseType)

WoodJoints<-data.frame(FailStress,Joint,Wood,CaseType)
WoodJoints
summary(WoodJoints)

#Now set the "effects sum to zero" convention for R output
options(contrasts = rep("contr.sum", 2))

#Here is some one-way analysis of the data (using r=9 case types)
plot(CaseType,FailStress,xlab="Case Type",ylab="Stress at Failure")
aggregate(WoodJoints$FailStress,by=list(WoodJoints$CaseType),mean)
aggregate(WoodJoints$FailStress,by=list(WoodJoints$CaseType),sd)

woodoneway.out<-lm(FailStress~CaseType,data=WoodJoints)
summary(woodoneway.out)
predict(woodoneway.out,se.fit=TRUE)
anova(woodoneway.out)

#Prepare what will be needed to make a plot of means with error bars
cellmeans<-aggregate(WoodJoints$FailStress,
                        by=list(WoodJoints$CaseType),mean)
cellmeans
cellmeans[,2]
sp<-sqrt(t(woodoneway.out$residuals)%*%
        woodoneway.out$residuals/woodoneway.out$df.residual)
sp
samplesizes<-c(2,1,2,2,2,2,2,2,1)
Delt<-rep(1,9)
Delt
for (i in 1:9) {Delt[i]<-qt(.975,
                      woodoneway.out$df.residual)*sp/sqrt(samplesizes[i])}
Delt

#Install the Hmisc Package and load it
#Load the stats package before proceeding and then make an interaction
#plot enhanced with error bars, much like Figure 8.3, page 550 of
#Vardeman and Jobe (that figure uses 99% error bars instead of 95% ones)

interaction.plot(Wood,Joint,FailStress,fun=mean,type="l",
ylim=c(0,3000),lw=2)
for(j in 0:2) {
  for(i in 1:3) {errbar(x=i,y=cellmeans[i+j*3,2],
                      yminus = cellmeans[i+j*3,2]-Delt[i+j*3],
                      yplus = cellmeans[i+j*3,2]+Delt[i+j*3],add=TRUE,errbar.col =j+2,lw=3)
    }
}

#Now do two-way analyses of the data
woodtwoway.out<-lm(FailStress~Joint+Wood+Joint:Wood,data=WoodJoints)
summary(woodtwoway.out)
predict(woodtwoway.out,se.fit=TRUE)
anova(woodtwoway.out)
woodtwoway.out2<-lm(FailStress~Wood+Joint+Joint:Wood,data=WoodJoints)
summary(woodtwoway.out2)
predict(woodtwoway.out2,se.fit=TRUE)
anova(woodtwoway.out2)

woodtwowaymain.out<-lm(FailStress~Joint+Wood,data=WoodJoints)
summary(woodtwowaymain.out)
predict(woodtwowaymain.out,se.fit=TRUE)
anova(woodtwowaymain.out)

#Here is a bit of R syntax that uses all main effects and interactions
#in a factorial analysis

summary(lm(FailStress~Joint*Wood))

#Here is the raw material for an F test of the hypothesis that there are
#no WoodxJoint Interactions

anova(lm(FailStress~Joint+Wood,data=WoodJoints),
      lm(FailStress~Joint+Wood+Joint:Wood,data=WoodJoints))

#Now set up a MLR analysis of the data

nWood<-as.numeric(Wood)
nJoint<-as.numeric(Joint)
W1<-rep(1,16)
for (i in 1:16) {if (nWood[i]==2) W1[i]<-0}
for (i in 1:16) {if (nWood[i]==3) W1[i]<--1}

W2<-rep(1,16)
for (i in 1:16) {if (nWood[i]==1) W2[i]<-0}
for (i in 1:16) {if (nWood[i]==3) W2[i]<--1}

J1<-rep(1,16)
for (i in 1:16) {if (nJoint[i]==2) J1[i]<-0}
for (i in 1:16) {if (nJoint[i]==3) J1[i]<--1}

J2<-rep(1,16)
for (i in 1:16) {if (nJoint[i]==1) J2[i]<-0}
for (i in 1:16) {if (nJoint[i]==3) J2[i]<--1}

W1J1<-W1*J1
W1J2<-W1*J2
W2J1<-W2*J1
W2J2<-W2*J2
cbind(CaseType,FailStress,W1,W2,J1,J2,W1J1,W1J2,W2J1,W2J2)

summary(lm(FailStress~W1+W2+J1+J2+W1J1+W1J2+W2J1+W2J2))
predict(lm(FailStress~W1+W2+J1+J2+W1J1+W1J2+W2J1+W2J2),se.fit=TRUE)
anova(lm(FailStress~W1+W2+J1+J2+W1J1+W1J2+W2J1+W2J2))
summary(lm(FailStress~J1+J2+W1+W2+W1J1+W1J2+W2J1+W2J2))
anova(lm(FailStress~J1+J2+W1+W2+W1J1+W1J2+W2J1+W2J2))

summary(lm(FailStress~W1+W2+J1+J2))
predict(lm(FailStress~W1+W2+J1+J2), se.fit=TRUE)
anova(lm(FailStress~W1+W2+J1+J2))

# Next are analyses for the (balanced data) 2x2x2 Metal Cutting Example
# that appears as Example 4 in Chapter 8 of Vardeman and Jobe
# First make a data frame

Tool<-rep(c(rep(1,4),rep(2,4)),4)
Angle<-rep(c(rep(1,8),rep(2,8)),2)
Cut<-c(rep(1,16),rep(2,16))
Power<-c(29.0,26.5,30.5,27.0,28.0,28.5,28.0,25.0,
         28.5,28.5,30.0,32.5,29.5,32.0,29.0,28.0,
         28.0,25.0,26.5,26.5,24.5,25.0,28.0,26.0,
         27.0,29.0,27.5,27.5,27.5,28.0,27.0,26.0)
Tool<-as.factor(Tool)
Angle<-as.factor(Angle)
Cut<-as.factor(Cut)
Miller<-data.frame(Power, Tool, Angle, Cut)

Miller.out<-lm(Power~Tool*Angle*Cut, data=Miller)
summary(Miller.out)
predict(Miller.out)
coefficients(Miller.out)
anova(Miller.out)

Miller.out2<-lm(Power~Angle+Cut,data=Miller)
summary(Miller.out2)
predict(Miller.out2)
coefficients(Miller.out2)
anova(Miller.out2)

#Now make some dummy variables that when used will produce the "all
#high" fitted effects, i.e. what is produced by the Yates algorithm

nTool<-as.numeric(Tool)
nAngle<-as.numeric(Angle)
nCut<-as.numeric(Cut)

T2<-rep(1,32)
A2<-rep(1,32)
C2<-rep(1,32)
for (i in 1:32) {if (nTool[i]==1) T2[i]<--1}
for (i in 1:32) {if (nAngle[i]==1) A2[i]<--1}
for (i in 1:32) {if (nCut[i]==1) C2[i]<--1}

T2A2<-T2*A2
T2C2<-T2*C2
A2C2 <- A2*C2
T2A2C2 <- T2*A2*C2
cbind(Power, T2, A2, C2, T2A2, T2C2, A2C2, T2A2C2)

Miller.out3 <- lm(Power ~ T2 + A2 + T2A2 + C2 + T2C2 + A2C2 + T2A2C2)
summary(Miller.out3)
predict(Miller.out3)
coefficients(Miller.out3)
anova(Miller.out3)

Miller.out4 <- lm(Power ~ A2 + C2)
summary(Miller.out4)
predict(Miller.out4)
coefficients(Miller.out4)
anova(Miller.out4)