1. **Read the data into R. Make a data frame, naming the variables with the names above. Print out the first 10 lines.** [5 marks]

The following code reads in the data, and names the variables. Note that there is no header line in the file.

```r
pbf.df = read.table("C:\Users\alee044\Documents\Teaching\330\assignments\2010\Assignment 3\PBF.txt",header=FALSE)
```

```r
names(pbf.df) = c("PBF", #Percent body fat using equation, 457/Density - 414.2
"Density", #Density (gm/cm^3)
"Age", # Age (yrs)
"Weight", # Weight (lbs)
"Height", # Height (inches)
"BMI", # Adiposity index = Weight/Height^2 (kg/m^2)
"Neck", # Neck circumference (cm)
"Chest", # Chest circumference (cm)
"Abdomen", # Abdomen circumference (cm)
"Hip", # Hip circumference (cm)
"Thigh", # Thigh circumference (cm)
"Knee", # Knee circumference (cm)
"Ankle", # Ankle circumference (cm)
"Biceps", # Extended biceps circumference (cm)
"Forearm", # Forearm circumference (cm)
"Wrist") # Wrist circumference (cm) "distal to the styloid processes"
```

Note that all characters after the # are ignored. The first 10 lines are

```r
> pbf.df[1:10,]
   PBF Density Age Weight Height BMI Neck Chest Abdomen Hip Thigh Knee Ankle Biceps Forearm Wrist
 1    12.6     1.0708  23 154.25  67.75 23.7 36.2  93.1    85.2  94.5  59.0 37.3  21.9   32.0    27.4  17.1
 2     6.9     1.0853  22 173.25  72.25 23.4 38.5  93.6    83.0  98.7  58.7 37.3  23.4   30.5    28.9  18.2
 3    24.6     1.0414  22 154.00  66.25 24.7 34.0  95.8    87.9  99.2  59.6 38.9  24.0   28.8   25.2  16.6
 4    10.9     1.0751  24 184.75  72.25 24.9 37.4 101.8    86.4 101.2  60.1 37.3  22.8   32.4   29.4  18.2
 5    27.8     1.0340  24 184.25  71.25 25.6 34.4  97.3 100.0 101.9  63.2 42.2  24.0   32.2  27.7  17.7
 6    20.6     1.0502  24 210.25  74.75 26.5 39.0 104.5  94.4 107.8  66.0 42.0  25.6   35.7  30.6  18.8
 7    19.0     1.0549  26 181.00  69.75 26.2 36.4 105.1  90.7 100.3  58.4 38.3  22.9  31.9  27.8  17.7
 8    12.8     1.0704  25 176.00  72.50 23.6 37.8  99.6  88.5  97.1  60.0 39.4  23.2  30.5  29.0  18.8
 9     5.1     1.0900  25 191.00  74.00 24.6 38.1 100.9  82.5  99.9  62.9 38.3  23.8  35.9  31.1  18.2
10   12.0     1.0722  23 198.25  73.50 25.8 42.1  99.6  88.6 104.1  63.1 41.7  25.0  35.6  30.0  19.2
```

2. **I have not changed any values in the original data set, but there are several strange values. Identify these using graphical methods and either correct them or delete the offending observations. (Delete a maximum of 4.) In particular, some of the PBF
values seem suspect (which ones?) Calculate the volume from the variables Density and Weight. [5 marks]

For a first look, we can calculate a pairs plot:

![Pairs plot](image)

Seems though there are a lot of outliers. To identify them, we can use the function `order`. This will give the index of the largest and smallest observations for a particular variable:

```r
> order(pbf.df$Weight)
```
The large weight visible in the pairs plot is observation 39. Similarly the small height is observation 42 (with a height of 29.5 inches and a weight of 205 pounds!), the large hip, thigh and knee all 39, the two large ankles 31 and 86.

Deleting these (and the last 2 observations as well) gives a better pairs plot:

```r
pairs(pbf.df[-c(31,39,42,86),])
```

There are some other points liable to have large influence, but we have used up our outlier budget. Let's work with this reduced data set, plus deleting the last 2 observations:

```r
pbf.use.df = pbf.df[-c(31,39,42,86,251,252),]
```

We will adjust the row labels to be 1 to 248:
To check out the accuracy of the PBF calculation, plot the calculated PBF against the density, using the index number as the plotting symbol:

![Graph showing PBF against density with points labeled 45, 73, 92, 178.]

Seems like 4 points are miscalculated, namely 45, 73, 92, 178.

Since these will not affect rest of the analysis, we won’t correct them.

To calculate the volume in litres and add this variable to the data frame, we type

```r
Volume = pbf.use.df$Weight*453.59237/ (1000*pbf.use.df$Density)
pbf.use.df = data.frame(pbf.use.df, Volume)
```

3. **Develop a model that will predict the volume from the other variables, excluding Density and PBF. You should be able to come up with a model that predicts very well. Points to note: Which variables should be selected? Are transformations indicated? (think Cherry trees). You should potentially consider using all the techniques you have been taught, up to the end of lecture 15. [20 marks]**

It seems as though the relationship between Volume and the other variables could well be multiplicative, as it was with the cherry trees. Let’s log all the variables, making new, logged variables. This will be necessary for the variable selection methods to work. We will eliminate Density and PBF as they are no longer needed.
Here is a quick way to do this (it works because all the variables are numeric)

```r
> log.pbf.df = log(pbf.use.df)[,-c(1,2)]
> newnames = paste("log.", names(pbf.use.df)[-c(1, 2)], sep="]
> newnames
> names(log.pbf.df)=newnames
```

There are substantial correlations between the variables, so not all will be needed. To figure out which ones are required, we do some variable selection. First all possible regressions:

```
rss  sigma2  adjRsq  Cp  AIC  BIC  CV log.Age
1 0.04933  2e-04  0.99244  326.3636  572.3636  579.37 43 0.00490       0
2 0.02322  1e-04  0.99643  273.4743  283.9951  293.78 59 0.00233       0
3 0.02179 13.05652  259.0545  273.0759  284.5963  294.39 86 0.00220       0
4 0.02129  9e-05  0.99670  254.4801  275.5120  285.31 45 0.00217       0
5 0.02105  8.48006 254.4801  275.5120  285.31 45 0.00217       0
6 0.02084  8.08625 254.4801  275.5120  285.31 45 0.00217       0
7 0.02064  7.81664 253.1565  274.7045  284.5963  294.39 86 0.00216       0
8 0.02041  7.15648 253.1565  274.7045  284.5963  294.39 86 0.00216       0
9 0.02026  7.43173 253.1565  274.7045  284.5963  294.39 86 0.00216       0
10 0.02018  7.15648 253.1565  274.7045  284.5963  294.39 86 0.00216       0
11 0.02011  7.15648 253.1565  274.7045  284.5963  294.39 86 0.00216       0
12 0.02009 11.42745 257.4275  293.0379  302.9968  312.95 88 0.00223       0
13 0.02007 13.23951 259.2395  308.3141  318.2921  328.27 02 0.00225       0
14 0.02005 15.00000 261.0000  313.5800  323.5579  333.53 13 0.00228       1
```


Stepwise regression gives
model.lm = lm(log.Volume~., data=log.pbf.df)
null.lm = lm(log.Volume~1, data=log.pbf.df)
step(null.lm, scope=formula(model.lm), direction="both")

> output not shown
Call:
lm(formula = log.Volume ~ log.Weight + log.Abdomen + log.Wrist +
    log.Height + log.Chest +

which is the same as the eight-variable model chosen by APR. Both the 4 and 8-variable models should be OK for prediction. In fact the 4 variable model has an $R^2$ almost as good as the 8-variable model, so we go with the 4-variable model in the interest of simplicity.

Let's subject these models to some checks:

> model.6.lm = lm(log.Volume~log.Age + log.Weight + log.Height +
> par(mfrow=c(2,2))
> plot(model.6.lm)

A hint of curvature is present, and the Boxcox plot (not shown) indicates squaring the response might be a good idea. This leads to the model with plots
which look good apart from the except for an outlier pt 92. This however does not seem to be affecting the coefficients too much, as the influence plot shows. (Cooks distance is OK). Cov ratio indicates point 92 is affecting standard errors. No big outliers. Model seems OK, could use it for prediction. We will explore the effect of point 92 in the predictions.

4. *I have replaced the values of the variables PBF and Density on the last two individuals in the data set with NA’s. Using your model, predict the body volume for these two individuals.* [10 marks]

Code for predictions (with and without the high CR point 92):

```r
# predictions
# 4.var model, all data
predict.df = log(pbf.df[251:252,-(1:2)])
names(predict.df) = names(log.pbf.df)[-15]
```
> exp(sqrt(predict(model.42.lm, predict.df, interval="p")))
  fit   lwr   upr
251 82.87071 81.31958 84.44463
252 90.54701 88.86330 92.25543

# without pt 92
lv2.no92=lv2[-92]
model.42.no92.lm = lm(lv2.no92~ log.Weight + log.Height +
  log.Abdomen + log.Wrist , data=log.pbf.df[-92,])

> exp(sqrt(predict(model.42.no92.lm, predict.df, interval="p")))
  fit   lwr   upr
251 82.92558 81.41133 84.46152
252 90.62210 88.97804 92.28970

The results are very similar. We will go with the last one.

NB: There will be a prize for the best predictions. In the event of a tie, a stochastic mechanism will be used.

Extra question for 762 students

Suppose we logged the volume and the other variables (excluding PBF and Density), and fitted a model to log volume, using the other logged variables. Can you explain why we would not need to include the variable log(BMI) in the model, given the other variables are included?

Since

\[ \log(BMI) = \log(\text{Weight}/\text{Height}^2) = \log(\text{Weight}) - 2\log(\text{Height}) \]

there is an exact linear relationship between the logged variables. Thus, if Weight and Height are in the model, adding BMI will not reduce the RSS at all (ie we have perfect collinearity.) If we did try to include it, the software would just ignore it.