Department of Statistics

COURSE STATS 330/762

Model Answers for Assignment 1, 2012

Question 1.

Instructions

I. *Load the data into R, and make a data frame **banknotes.df** to contain the data. Check for any typographical errors (the data on the question sheet may be taken to be the correct data, but the data on the web may have been corrupted). Print out the last 10 lines of the data file. [5 marks]*

The following code will read in the data, using the web address syntax:

```
banknotes.df =
read.table("http://www.stat.auckland.ac.nz/~lee/330/
          datasets.dir/banknotes.txt")
```

We have omitted **header=TRUE** to cope with the row labels in the data file. You can check this file most easily by a pairs plot – this should make any outliers stand out:

```
> pairs(banknotes.df[,-7])
```

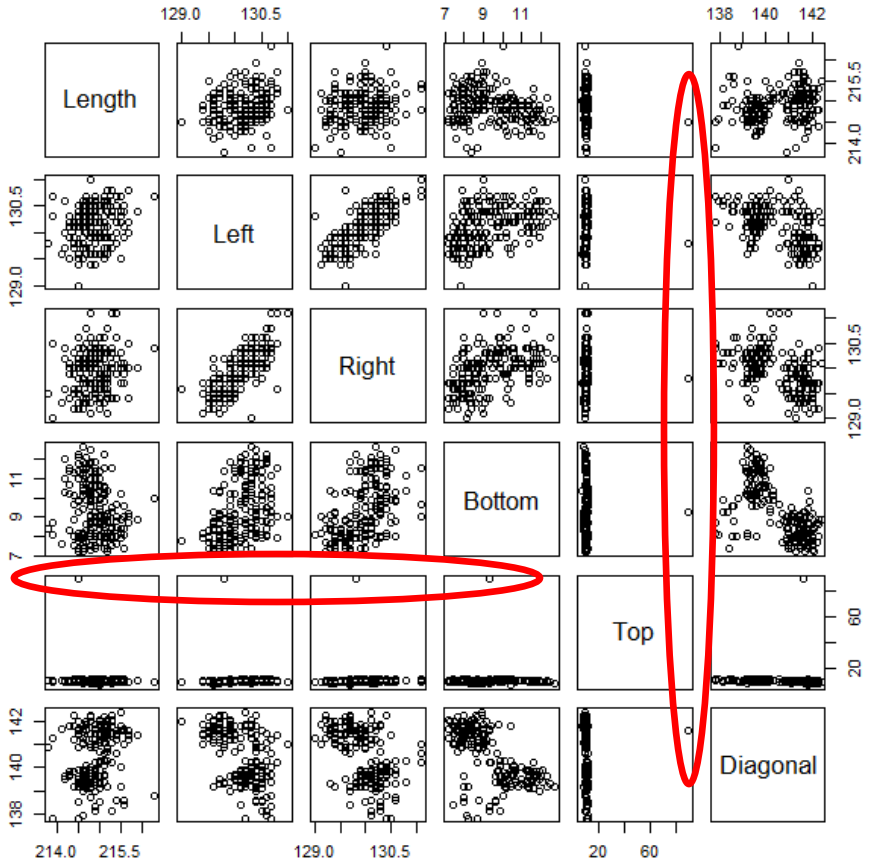
The plot is shown at the top of the next page. It is clear that there is a large value for the variable Top: ordering the values allows us to see which one:

```
> order(banknotes.df$Top)[200]
[1] 16
```

Note that the code **order(banknotes.df$Top)** prints the positions of the smallest value of Top, the next smallest, and so on up to the largest. Since there are 200 values, the 200th element of the vector calcuated by **order(banknotes.df$Top)** gives us the position of the largest value, the error; i.e. the error is in line 16 of the data file:

```
> banknotes.df[16,]
   Length  Left Right Bottom  Top Diagonal Y
16  214.5 129.8 129.8    9.3 88.5    141.6 0
```

The bad value is 88.5. Inspecting the data sheet, we see it should be 8.5.



To correct it, type

```
> banknotes.df[16,5] = 8.5
```

The last 10 lines can be printed by

```
> banknotes.df[191:200,]
    Length  Left Right Bottom  Top Diagonal Y
191  215.1 130.2 129.8   10.2 12.0    139.4 1
192  215.4 130.5 130.6    8.8 11.0    138.6 1
193  214.7 130.3 130.2   10.8 11.1    139.2 1
194  215.0 130.5 130.3    9.6 11.0    138.5 1
195  214.9 130.3 130.5   11.6 10.6    139.8 1
196  215.0 130.4 130.3    9.9 12.1    139.6 1
197  215.1 130.3 129.9   10.3 11.5    139.7 1
198  214.8 130.3 130.4   10.6 11.1    140.0 1
199  214.7 130.7 130.8   11.2 11.2    139.4 1
200  214.3 129.9 129.9   10.2 11.5    139.6 1
```

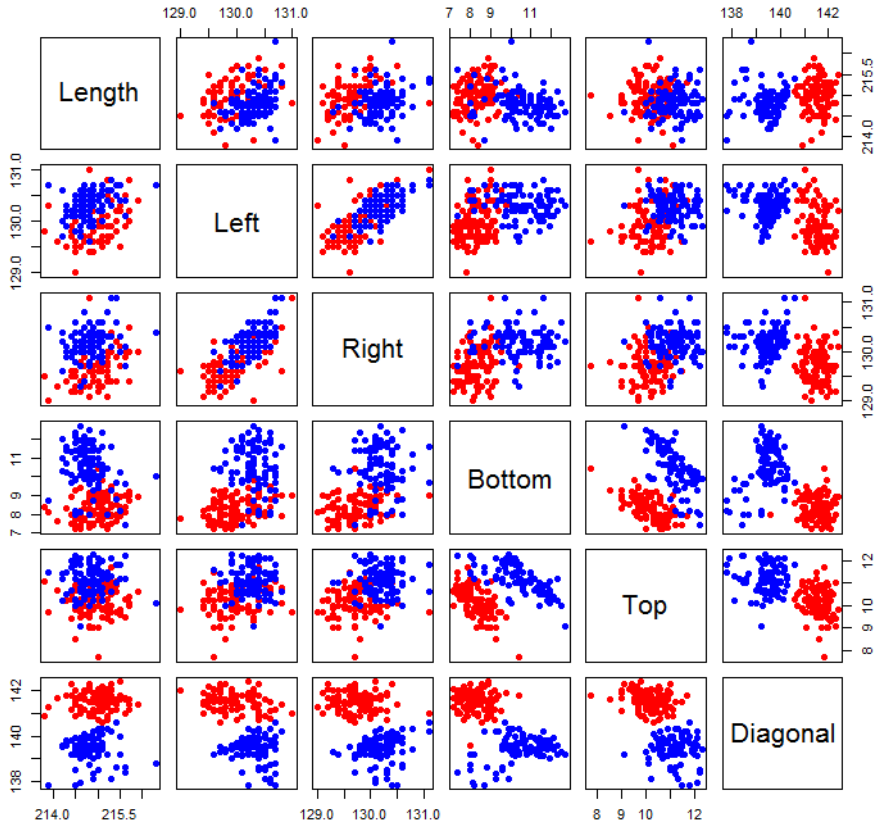
[5 marks: 2 for reading 1 for finding the mistake, 1 for correcting it, 1 for printing.]

I. *Using a suitable plot or plots, devise a graphical method that will allow you to discriminate between genuine and counterfeit banknotes on the basis of these measurements. Can you describe the method in words so that a non-numerate person could understand it? [10 marks]*

The pairs plots used to detect any erroneous values seemed to show two clusters of points. Perhaps these correspond to the genuine and counterfeit banknotes? We can check this by plotting the points using different colours, say red for genuine and blue for countertfeit. We can do this with the pairs function:

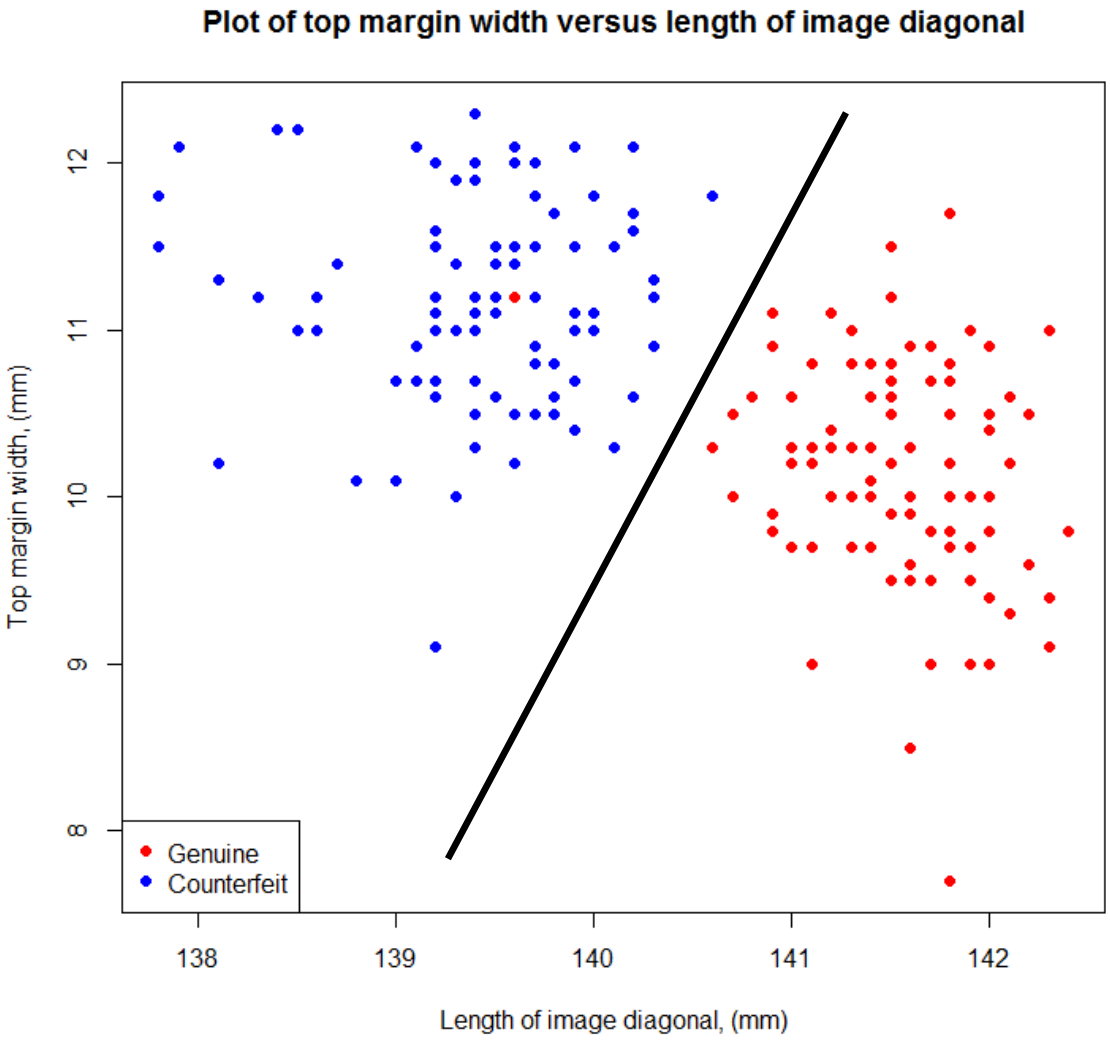
```
mycol=c("red","blue")
pairs(banknotes.df[,-7], col=mycol[banknotes.df$Y+1], pch=19)
```

This works because **`banknotes.df$Y+1`** ia a vector of length 200, having value 1 for genuine banknotes and 2 for counterfeits. The argument **`col`** controls the colour of the individual points. The argument **`pch=19`** uses solid circles as plotting symbols (the default is **`pch=21`** which is an open circle; these are harder to see.)



There is excellent separation between the two clusters in the plot of Diagonal versus Top: let's replot it on a bigger scale:

```
plot(banknotes.df$Diagonal, banknotes.df$Top, type="n",xlab = "Length
of image diagonal, (mm)",
ylab = "Top margin width, (mm)",
main = "Plot of top margin width versus length of image diagonal")
points(banknotes.df$Diagonal, banknotes.df$Top,
col=mycol[banknote$Y+1], pch=19)
legend(x="bottomleft", legend = c("Genuine","Counterfeit"), col=mycol,
pch=19)
```



Apart from the isolated red point (was this misclassified in the original data set?) there is excellent separation. Note that the bold separating line was not drawn by R, I added it using MS Word features.

To discriminate between genuine and counterfeit banknotes, we plot the point corresponding to the new banknote on the graph above. If it is on the ``red'' side of the line, we declare it to be genuine, otherwise counterfeit.

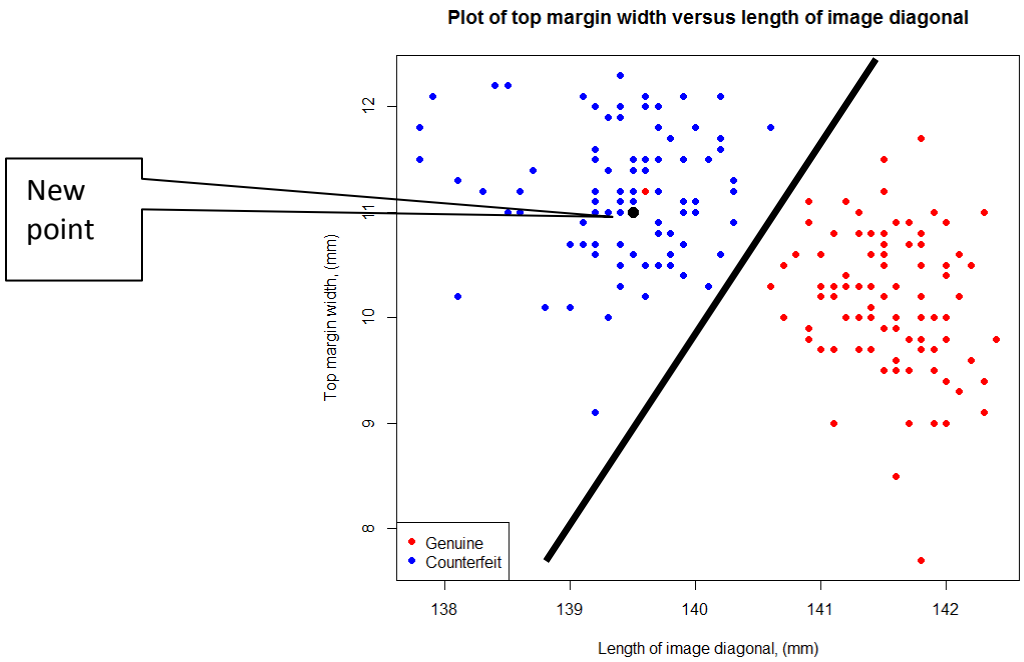
[10 marks: most students will draw a plot similar to the one above, using two variables to discriminate. Deduct 4 marks if they use only one variable at a time (using boxplots say). Deduct up to 3 marks if they don't give a clear explanation of how a new banknote would be classified. They may use another pair of variables.]

3. Suppose a banknote has Diagonal = 139.5mm; Right = 130.0mm; Top=11.0mm. Is it genuine or a fake? Why?[5 marks]

We can add the point (as a larger black dot, using the argument cex=1.5) to the plot using the code

```
> D = 139.5; R = 130.0; T=11.0
> points(D, T, pch=19, cex=1.5)
```

The result is shown below. It is clear that the banknote is very likely to be counterfeit, as it falls on the ``counterfeit'' side of the line.



Question 2

1. *Load the data into R.* and print it out to three decimal places. [3 marks]

```
> plot.df = read.table(
"http://www.stat.auckland.ac.nz/~lee/330/datasets.dir/plot.csv",
sep=",")

>round(plot.df,3)
```

2. *Draw four plots (on the same page) of the data that resemble as much as possible the figure below. Pay careful attention to the labeling of the points and the axes. [12 marks]*

Required features of the plot are

I. Labeled axes, going from 0.0 to 1 and 3 to 7 (use **xlim**, **ylim**)
II. Different line styles and thicknesses (**lty** and **lwd** arguments to **plot**)
III. Different coloured lines (black, grey)
IV. Text above and below plotted points (**pos** argument to **text**)
V. Plotting points and lines (**type="b")**
VI. Four plots to a page (**par(mfrow=c(2,2))**

The following code implements these.

```
par(mfrow=c(2,2))
plot(y~x, type="n", xlim=c(0,1), ylim=c(3,7), main =
"Solid",data=plot.df)
lines(plot.df$x,plot.df$y, lwd=2)
#
plot(y~x, type="n", xlim=c(0,1), ylim=c(3,7), main =
"Dashed",data=plot.df)
lines(plot.df$x,plot.df$y, lty=2,lwd=2)
#
plot(y~x, type="l", xlim=c(0,1), ylim=c(3,7),
    col="darkgrey", main ="Grey",data=plot.df)
#
plot(y~x, type="b", xlim=c(0,1), ylim=c(3,7), main
="Numbered",data=plot.df)
mypos = c(3,1,3,3,3,3,1,3,3,1)
text(plot.df$x,plot.df$y, row.names(plot.df), pos=mypos)
```

[12 marks] : 2 marks for each of the features I-VI successfully implemented.

3. *The x-values in the file are in ascending order. If this had not been the case what code would be necessary to allow for this? [3 marks]*

   The code above will not work unless the x-values are in increasing order. This is because when we do a plot with type="l" or type="b", the first point in the file is joined to the second, the second to the third and so on, Try it and see what happens if the x-values are not increasing – see the rat example in Lecture 2.

   To fix the problem we need to sort the rows so that the x's increase. We need to make sure the y-values are not mixed up. The easiest way to do this is to use the order function:

```
> x.order = order(plot.df$x)
> plot.df = plot.df[x.order,]
```

   Then we can proceed as before. [3 marks]