In [1]:
```
# A thing to adjust the size of the plot
options(repr.plot.width=5, repr.plot.height=4)
```

Question 1: Breaking Dataset

By Musa Rasheed (25618232)

An experiment was conducted to select the supplier of raw materials for production of a component. The breaking strength of the component (column breaking) was the objective of interest. 4 suppliers were considered (column supplier). The four operators (column operator) can only produce one component each per day.

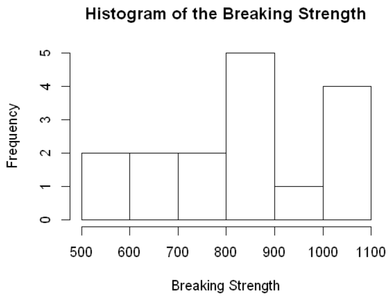
Let's begin by loading in the data from the text file:

In [2]:
```
broke = read.table('breaking.txt', header = TRUE, sep = "", dec = ".")
broke
```

breaking	operator	day	supplier
810	op1	day1	B
1080	op1	day2	C
700	op1	day3	A
910	op1	day4	D
1100	op2	day1	C
880	op2	day2	D
780	op2	day3	B
600	op2	day4	A
840	op3	day1	D
540	op3	day2	A
1055	op3	day3	C
830	op3	day4	B
650	op4	day1	A
740	op4	day2	B
1025	op4	day3	D
900	op4	day4	C

A) Create a histogram of breaking strength and label the axes properly. Comment on the shape of the distribution. (3 marks)

In [3]:
```
hist(broke$breaking,
     main = "Histogram of the Breaking Strength",
     xlab = "Breaking Strength",
     ylab = "Frequency")
```



The breaking strength histogram seems to be fairly flat up until the 800-1100 range. There are 2 distinct peaks that make this a bimodal histogram (although this could be due to a lack of data as there are only 16 entries).

B) Find the five number summary and the standard deviation for the variable breaking strength. (2 marks)

This can be done by simply using the `summary()` function:

In [4]:
```
summary(broke$breaking)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  540.0   730.0   835.0   840.0   938.8  1100.0 
```

The 5 number summary is shown above through R (simply ignore the mean column, or see below).

```
    Min      Q1  Median     3rd Qu.    Max. 
  540.0   730.0   835.0     938.8    1100.0 
```

Now to find the standard deviation, we can use the `sd()` function:

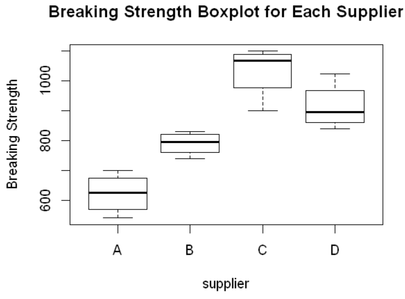
In [5]:
```
sd(broke$breaking)
```

170.029409220876

C) Create a proper graphical display to compare the breaking strength distributions of the four different suppliers and label the axes properly. Which supplier would you recommend? Justify your answer with a brief sentence. (3 marks)

Inspiered by the last question, I will use side-by-side boxplots (similair to what is done in Lab 1 Q4).

In [6]:
```
boxplot(breaking~supplier,
        data = broke,
        main = "Breaking Strength Boxplot for Each Supplier",
        ylab = 'Breaking Strength')
```



Assuming the objective of the raw material is to have the highest breaking strength possible, supplier C seems to be the best choice given that it's median breaking strength is by far the highest indicating a relativley consistent result compared to the others.

(Of course we may need more data to make a more concrete decision given that there are only 4 entries per supplier)

D) Suppose the breaking strength for operator 3 on day 2 was entered incorrectly and should have been 600 instead of 540. How would your answer in part (b) change? (2 marks)

To answer this, we simply need to change the data. There are a number of ways to make this change, but I will use a siply approach. I'll simplly call the row and column that data is in and change in manually.

In [7]:
```
broke$breaking[10] = 600
broke[10,]
```

	breaking	operator	day	supplier
10	600	op3	day2	A

In [8]:
```
#Now we can reuse the summary function from before on the modifed dataset

summary(broke$breaking)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  600.0   730.0   835.0   843.8   938.8  1100.0 
```

This is the new 5 number summary:

```
    Min      Q1  Median     3rd Qu.    Max. 
  600.0   730.0   835.0     938.8    1100.0 
```

Since the quartiles and median are based on position in the ordered list (when we organize from smallest to largest) the summary does not change all that much. Since 600 is the lowest number in the group, it beccomes the new minimum leaving the rest of the numbers untouched.

As for the standard deviation:

In [9]:
```
sd(broke$breaking)
```

163.508409569661

It has changed from 170.029409220876 to 163.508409569661. This smaller standard deviation indicates that the values are more similair in value to eachother (whch they are since the minimum was raised, thus bringing it closer to the other values)