**Stat 407 Lab 2 Summary Statistics of Multivariate Data**
**Due: Wednesday, August 29 2012, in class.** Hand in one solution per group.

**Purpose:** To discuss the use of summary statistics for multivariate data, and to learn how to use software to generate summary statistics.

The final medal tally for the 2012 Olympic Games is given in the file `olympics2012.csv` on the class web site, for you to download and read into R. It contains the variables on the number of gold, silver and bronze medals for each country and the population of each country. Data was taken from `http://www.london2012.com` and the population data from `http://www.wikipedia.org`.

**Instructions:**

- You'll need to download the data from the course web site, and read it into R. Remember the `read.csv`. Sometimes you need to use the input option `row.names=` to effectively ignore the first column, but you shouldn't need it this time. You can also set any of the columns with unique values as the row names after reading it on. Call your data.frame `olympics` to match the rest of my sample code.

```
>  olympics<-read.csv(file.choose()) # read in olympics2012.csv
>  rownames(olympics)<-olympics$Code
```

- Remember the `str` command for finding the size of the data, and the `head` command to look at the top of the data frame. Another command that is useful is `dim` which also gives the size of the data.

```
> head(olympics)
> str(olympics)
> dim(olympics)
```

- To round numbers, or get nice output, you can set the R options, or use `round` or `signif` functions.

```
> round(mean(olympics$Gold), 2)
> signif(mean(olympics$Gold), 5)
> options(digits=2)
```

- To generate the basic summary statistics, use the `summary` function. Alternatively, another useful way to generate summary statistics is to use the `apply` function to run statistics function on either the rows or the columns of the data matrix.

```
> summary(olympics)
> apply(olympics[,4:7], 2, mean) # Calculate the mean of the columns of the data
> apply(olympics[,4:7], 2, sd) # Calculate the standard deviation of the
   columns of the data
> apply(olympics[,4:6], 1, sum) # Sum the number of gold, silver, bronze
```

```
medals for each row
> apply(olympics[,4:7], 2, sum)/300 # Calculate the mean of the columns,
   by first computing the sum, and then dividing by the number of observations
> apply(olympics[,9:11],2,sum) # Adds up the number of medals given to men and women.
```

- To add rows to a data.frame, column by column, you can use the `data.frame` function to create a new data frame.

```
> olympics.sub<-data.frame(gold=c(olympics[,4], rep(0,119)),
    silver=c(olympics[,5], rep(0,119)), bronze=c(olympics[,6], rep(0,119)),
    total=c(olympics[,7], rep(0,119)))
```

- The `subset` function is useful for pulling out certain parts of the data.

```
> olympics.morethan10<-subset(olympics, Total>=10) #Countries with 10 or more medals.
```

- Use the `cor` function to calculate the correlation.

```
> cor(olympics.sub)
```

- To make a new variable in the data frame, remember the `$` operator. You can also use the `sort` and `order` functions to write out results depending on the magnitude of the numbers.

```
> olympics$percapita <- olympics$Total/olympics$Population*1000000
> olympics[order(olympics$percapita, decreasing=T),]
```

- To calculate the distance between two rows of a data frame, use the `dist` function.

```
> dist(olympics[1:2,4:6])
> dist(olympics.morethan10[,4:6])
> as.dist((1 - cor(t(olympics.morethan10[,4:6])))/2)
> olympics.morethan10[c(1,3,5,2,4),4:6] $ Print out a few countries
```

- **Points will be awarded for nice formatting of your answers to these questions, and appropriate rounding of numbers.**

**Exercises:**

1. How many countries are represented in the data, ie what is $n$?
2. What would you consider to be the dimension of the data, $p$?
3. How many gold, silver and bronze medals were awarded? Should there be an equal number? If not, why do you think they might differ?

4. Use the web to find out how many countries participated, and hence how many countries didn't win a medal.

5. Calculate the means and standard deviations for the total medal counts? (Four variables: Gold, Silver, Bronze, Total).

6. What's wrong with the calculations in the previous questions? (Hint: Refer to Q4.) Recompute these numbers correctly.

7. Compute the correlation matrix for the gold, silver, bronze medal counts. Explain what this tells about this data? (Hint: Are countries that win gold also likely to win silver and bronze?)

8. Compute the total number of medals per million people for each country. Which are the top five ranked countries by this measure?

9. How many total medals awarded to men? women? Why do you think that the total for men is higher than that for women?

10. Compute the Euclidean distance between China and the USA, on medal counts for gold, silver, and bronze. (Calculate by hand, and with R to double-check the answer.)

11. For countries with 10 or more total medals, calculate the Euclidean distances with USA. Which country is the most similar to USA? (Smallest distance)

12. Let's look at similarity in a different way. Suppose that we don't care about overall magnitude of the number of medals, but rather how similar in relative counts the countries are. We could this by designing a distance measure based on correlation:

$$d_{Cor}(x, y) = (1 - r)/2, \qquad r = cor(x, y)$$

    Confirm that this is indeed a distance metric.

    With this distance, which country is the most similar to the USA? Use only the countries with at least 10 medals. What is the pattern that both countries share in terms of their medal distribution? How is this different from the distribution of medals for China?

13. Generally, for any multivariate data set, explain in English what you learn by examining the summary statistics of multivariate data. Why is it important to calculate these and study them? What type of information can't you learn about the nature of and relationship between variables, from studying only the mean and variance-covariance or correlation? (Hint: Summary statistics tell you about center and spread, but not much detail about the distribution of values.)