StatR 101: Fall 2012
Homework 1 - Solutions
Eli Gurarie
Due Saturday, October 6, 11:55 pm

3. Create a vector of all the even numbers from 2 to 100 using the constraints given below:

```
a) seq(2,100,2)
b) 1:50*2
c) cumsum(rep(2,50))
```

4. Exact answers should be different for each individual because the data were generated randomly.

```
a) names(Grades) <- Names
b) Grades[Grades >= 90]
c) Grades[Grades < 60]
d) Grades.F <- Grades[Sex =="F"]
   Grades.M <- Grades[Sex =="M"]
e) MeanGrade.M <-sum(Grades.M)/length(Grades.M)
   MeanGrade.F <-sum(Grades.F)/length(Grades.F)
```

5. More grade manipulation...

(a) Create a vector for each of the tests. The vectors should contain the 5 student scores for that test (i.e. quiz1,quiz2, midterm, final) as shown in Table 1.

```
quiz.1 <-c(89,95,75,82,90)
quiz.2 <-c(87,90,60,0,90)
midterm <-c(84, 92, 72, 88, 96)
final <-c(78,76, 58, 68, 80)
```

(b) Read about the `cbind()` and `rbind()` functions. Use one of them to create a matrix with the test scores reported in rows.

```
ClassScores <- rbind(quiz.1, quiz.2, midterm, final)
```

(c) Name the columns of this matrix (using the `colnames()` function) to reflect the column names in Table 1.

```
colnames(ClassScores) <- c("Alice","Bruno","Carl","Dolores","Ebenezer")
```

(d) Suppose Alice wants a list of her test scores in the class. Create a vector using the matrix ClassScores with the test scores for only Alice, call it `AliceProgress`.

```
 AliceProgress <- ClassScores[,1]
```

(e) Create a new vector using the matrix `ClassScores` that contains the average quiz scores for each student (average for quiz 1 and quiz 2), call it `QuizAverage`.

```
 QuizAverage <- (ClassScores[1,]+ClassScores[2,])
# or:
 QuizAverage <- colMeans(ClassScores[1:2,])
```

(f) Final grades in this class are calculated as follows: quizzes 20%, midterm 25%, final 55%. Use the matrix `ClassScores2` to create a vector that contains the students overall scores in the class, call it `FinalGrades`.

```
# Most of you did this, which is perfectly fine:
 FinalGrades <- 0.55*ClassScores2[4,]+ 0.25*ClassScores2[3,] +
                      0.2*ClassScores2 + [5,]FinalGrades
# Here is a slightly more "generic" way to do it, i.e. if the weights
# change, it is easier to compute.
 weights <- c(0.25, 0.55, 0.2)
 weights.Matrix <- matrix(weights, ncol=5, nrow=3)
 weighted.Grades <- weights.Matrix * ClassScores2[3:5,]
```

(g) Use the `mean()` and `sd()` function in R to calculate the mean and standard deviation for the final grades.

```
mean(FinalGrades)
sd(FinalGrades)
```

Bonus problem: Later in the quarter we will discuss standard deviations in detail, including the source of the difference between the two formulas. In the meantime:

(a) The formula that `R` uses is

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x-\overline{x})^2} \tag{1}$$

which is the *sample standard deviation.* It is an *estimate* of the variance of a population which is represented by a random sample.

(b) Because we have access to the entire population (in this case, all the students in the class) we should use the population standard deviation (which the Wikipedia article calls "the standard deviation of the sample"):

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x-\overline{x})^2} \tag{2}$$

In `R`:

```
(1/length(FinalGrades)) * sum((FinalGrades - mean(FinalGrades))^2)
```

This number will always be a bit smaller than the sample standard deviation, because the numerator is larger.