BMI 713: Computational Statistics for Biomedical Sciences

Assignment 4

September 30, 2010 (due October 7)

Loops and Functions in R

1. Let `M` be the following 4×4 matrix: `M = matrix(1:16, ncol=4)`. We would like to compute the sum of all the elements in the matrix.

   (a) Use a double loop to go through each element of the matrix, adding one number at a time.

   ```
   > s <- 0
   > for (i in 1:nrow(M))
   +     for (j in 1:ncol(M))
   +         s <- s + M[i, j]
   > s
   [1] 136
   ```

   (b) Use a single loop, using a single command to sum up each row or column.

   ```
   > s <- 0
   > for (i in 1:nrow(M))
   +     s <- s + sum(M[i,])
   > s
   [1] 136
   ```

   (c) Use a single command to find the sum of the matrix.

   ```
   > sum(M)
   [1] 136
   ```

2. In the following problems, do not use any standard `R` functions.

   (a) Write a function that takes a single number and returns double that value.

   ```
   > double <- function(x) 2*x
   > double(3)
   [1] 6
   ```

   (b) Write a function which takes two numbers and returns their sum.

   ```
   > add <- function(a, b) a + b
   > add(3^2, 4^2)
   [1] 25
   ```

   (c) Write a function which takes a vector of values and returns its variance.

   ```
   variance <- function(v) {
       m <- 0
       for (x in v)
           m <- m + x
       m <- m / length(v)

       d <- (v - m)^2
       ss <- 0
       for (x in d)
           ss <- ss + x
   ```

```
    ss / (length(v) - 1)
}

# The variance function can be written quite succinctly
# if we do not disallow the use of sum and mean.
function(v) sum((v - mean(v))^2) / (length(v)-1)
```

(d) Write a function which takes a single number x and an integer n and returns an array with n copies of x. If the user does not specify n, use a default value of 2.

```
> dup <- function(x, n=2) {
+     result <- c()
+     for (i in 1:n)
+         result <- c(result, x)
+     result
+ }
> dup(2, 3)
[1] 2 2 2
> dup(2)
[1] 2 2

# Alternatively, a technique called recursion can be used.
> dup <- function(x, n=2) {
+     if (n > 1)
+         c(x, dup(x, n - 1))  # dup calls itself again for n > 1
+     else
+         x                    # dup does not call itself again for n <= 1
+ }
> dup(2)
[1] 2 2
> dup(2, n=10)
 [1] 2 2 2 2 2 2 2 2 2 2
```

(e) Write a function which takes a vector of numbers and determines whether the vector is a *palindrome*. A vector v is a palindrome if v is equal to the reverse of v. The return value of this function should be `TRUE` if the supplied vector is a palindrome and `FALSE` otherwise.

```
> is.palindrome <- function(v) {
+     b <- v == v[length(v):1]
+     for (x in b)
+         if (x == FALSE)
+             return(FALSE)
+     TRUE
+ }
> is.palindrome(c(2,3,2))
[1] TRUE
> is.palindrome(c(2,3,3))
[1] FALSE
> is.palindrome(c(2,3,3,2))
[1] TRUE

# Again, without the restrictions imposed by the homework,
# the natural way to write this in R is
> is.palindrome <- function(v) all(v == v[length(v):1])
```

Nonparametric test

3. An experiment on learning in animals measures how long it takes a mouse to find its way through a maze. The mean time is 18 seconds for one particular maze. A researcher thinks that a loud noise will cause the mice to complete the maze slower. She measures how long each of 10 mice takes with and without a noise as stimulus. The results are as in Table 1, where the measurements are in seconds.

Table 1: Time used for mice to complete the maze

mouse	1	2	3	4	5	6	7	8	9	10
Without noise	83.7	80.6	101.5	94.6	76.9	83.1	98.5	98.8	91.1	100.3
With noise	83.6	81.1	102.1	95.5	76.5	83.4	99.3	98.6	92.3	100.6

(a) Calculate the Wilcoxon signed-rank test statistic T.

Sol'n. *First we compute the vector of differences Z:*

$$Z = (0.1, -0.5, -0.6, -0.9, 0.4, -0.3, -0.8, 0.2, -1.2, -0.3).$$

Now we rank the $|Z_i|$:

$$R = (1, 6, 7, 9, 5, 3.5, 8, 2, 10, 3.5).$$

The sum of the ranks R_i whose $Z_i > 0$ is the Wilcoxon signed-rank statistic:

$$T = 1 + 5 + 2 = 8.$$

(b) What is the expected mean μ_T and standard deviation σ_T of T under the the null hypothesis?

Sol'n. *The distribution of the signed-rank statistic on n data points can be approximated by a normal distribution with mean*

$$\mu_T = \frac{n(n+1)}{4} = 27.5$$

and standard deviation

$$\sigma_T = \sqrt{\frac{n(n+1)(2n+1)}{24}} \approx 9.81.$$

(c) Perform the Wilcoxon signed-rank test and report the P-value. Is the test significant?

Sol'n. *We will derive both the exact p-value and the approximated p-value from the normal distribution described in part (b).*

To determine the exact p-value, we must consider each of the $2^{10} = 1,024$ possible ways to assign plus and minus signs to 10 ranks. The significance of the observed statistic is given by

$$p = \frac{(\#\textit{of sign assignments whose rank sum } T \leq 8)}{(\textit{total \# of sign assignments})}$$

Suppose we assign plus signs to ranks 1, 2 and 4 (and, implicitly, minus signs to every other rank). Then the signed-rank statistic is $1 + 2 + 4 = 7$, which is more extreme than the statistic we computed from the data. The following table shows how many sign assignments exist whose sum is less than or equal to 8.

# of + signs	*Rank sums $\leq$ 8*	*#*
0	*No plus signs assigned: $T = 0$*	*1*
1	$\{1, 2, \ldots, 8\}$	*8*
2	$\{1+2, 1+3, \ldots, 1+7, 2+3, 2+4, \ldots, 2+6, 3+4, 3+5\}$	*12*
3	$\{1+2+3, 1+2+4, 1+2+5, 1+3+4\}$	*4*
4	*No 4 numbers between 1 and 10 sum to $\leq$ 8.*	*0*
Total		*25*

Thus, the (one-sided) probability of seeing a more extreme statistic than $T = 8$ is $p = 25/1024 \approx 0.0244$.

We can also approximate T by a normal distribution with mean 27.5 and standard deviation 9.81. To compute the significance of the value 8 in a normal distribution $N(27.5, 9.81^2)$, use

```
> pnorm(8, mean=27.5, sd=9.81)
[1] 0.0234187
```

As we can see, the two p-values are quite similar and both would prompt one to reject H_0 at significance level $\alpha = 0.05$.

To perform the test in R:

```
> wilcox.test(x, y, paired=TRUE, alternative="less")

Wilcoxon signed rank test

data:  x and y
V = 8, p-value = 0.02441
alternative hypothesis: true location shift is less than 0
```

(d) Perform an appropriate t-test. Do you get the same conclusion based on the t-test and the Wilcoxon signed-rank test?

Sol'n. *The p-value from the t-test is slightly smaller than the p-value from the non-parametric test.*

```
> x
 [1]  83.7  80.6 101.5  94.6  76.9  83.1  98.5  98.8  91.1 100.3
> y
 [1]  83.6  81.1 102.1  95.5  76.5  83.4  99.3  98.6  92.3 100.6
> t.test(x, y, paired=TRUE, alternative="less")

Paired t-test

data:  x and y
t = -2.4038, df = 9, p-value = 0.01982
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
        -Inf -0.09259347
sample estimates:
mean of the differences
                  -0.39
```

4. Little et. al. [1] recently studied the discrimination of healthy people from those with Parkinson's disease (PD) based on a range of biomedical voice measurements. Each row in the table corresponds to one of 195 voice recording. Columns of the table correspond to different attributes of the voice recording. The column named *status* indicates the health `status` of the individual (1 – Parkinson's; 0 – healthy). In this assignment, we are mainly interested in the measurement *recurrence period density entropy* (`RPDE`, the 19th column).

   Read the data file *parkinson.data* into R and save it as a data frame.

```
url <- "http://archive.ics.uci.edu/ml/machine-learning-databases/parkinsons/parkinsons.data"
parkinson <- read.table(url, sep=",", header=T)
```

   (a) Split the data frame `parkinson` into two data frames according to the value of `status` and save them into two data frames, `p` (`status = 1`) and `h` (`status = 0`) (Do not print out these data frames.)

Sol'n. *The `$` operator can be used to select columns from a data frame by name. For example, `parkinson$status` is equivalent to `parkinson[, "status"]` which is equivalent to `parkinson[, 18]` (since `status` is the 18^{th} column in the data frame).*

```
p <- parkinson[parkinson$status == 1, ]  # Parkinson's patients
h <- parkinson[parkinson$status == 0, ]  # Healthy patients
```

(b) Extract the measurements `RPDE` from the data frames `p` and `h` and save them into two vectors `v.p` and `v.h`.

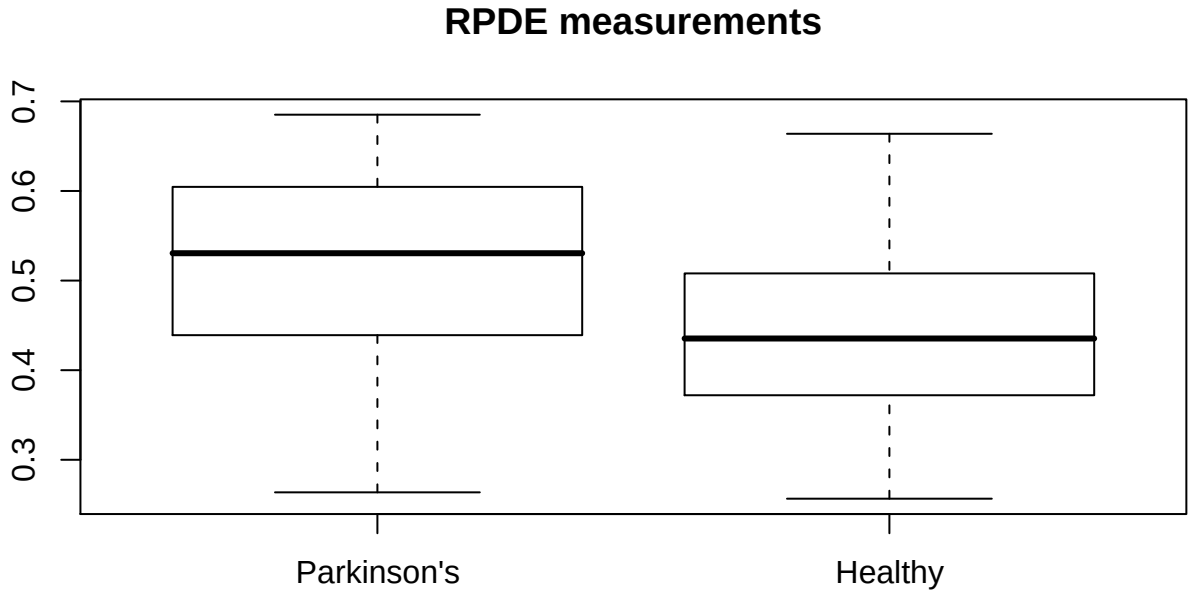
Sol'n. *Again, to extract a column we may use the `$` operator or select the column by number.*

```
v.p <- p$RPDE   # Equivalently, v.p <- p[ ,18]
v.h <- h$RPDE   # Equivalently, v.h <- h[ ,18]
```

(c) Plot the boxplots of `v.p` and `v.h` in the same plot.

Sol'n. *The `names` parameter to `boxplot` plots the group names along the bottom; `main` is the main title for the plot.*

```
boxplot(v.p, v.h, names=c("Parkinson's", "Healthy"), main="RPDE measurements")
```



(d) Perform Wilcoxon rank sum test to test if the median of `v.p` and `v.h` are the same. What is the value of the test statistic? What is the P-value?

Sol'n. *Use R's `wilcox.test` with `paired=FALSE` to perform a rank-sum test.*

```
> wilcox.test(v.p, v.h, paired=FALSE)

Wilcoxon rank sum test with continuity correction

data:  v.p and v.h
W = 4990, p-value = 0.00001669
alternative hypothesis: true location shift is not equal to 0
```

The test statistic is $W = 4990$ and the p-value is $p \approx 0.00017$. We reject the null hypothesis. median.

(e) Perform an appropriate t-test to test if the means of `RPDE` of the two groups are the same. Do you get the same conclusion?

Sol'n. *We perform a t-test without assuming equal variances.*

```
> t.test(v.p, v.h, paired=FALSE)

Welch Two Sample t-test

data:  v.p and v.h
t = 4.7268, df = 86.967, p-value = 0.000008717
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.04303617 0.10549192
sample estimates:
mean of x mean of y
0.5168159 0.4425519
```

Again. we reject the null hypothesis.

5. (Inspired by an example in [2]) Twelve sets of identical twins underwent psychological tests to measure the amount of aggressiveness in each persons's personality. We are interested in comparing the twins to each other to see if first born twin tends to be more aggressive than the other. The results are as in Table 2, the higher score indicates more aggressiveness.

Table 2: Aggressiveness of the twin

first born X_i:	86	71	77	68	91	72	77	91	70	71	88	87
second twin Y_i:	86	77	76	64	96	72	65	90	65	80	81	72

(a) State the null hypothesis and the alternative hypothesis.

Sol'n. *In this case, we will perform a signed-rank test since the data are paired. Thus, in the following hypothesis, μ refers to the mean* difference *between groups. $H_0 : \mu = 0$ and $H_A : \mu \neq 0$.*

(b) If we want to perform a nonparametric test, which one should be used, the Wilcoxon signed-rank test or the Wilcoxon rank sum test?

Sol'n. *Since the data are paired, we will treat the data as a single group of* differences. *The non-parametric analogue to the one-group t-test is the signed-rank test.*

(c) Perform the test, report the test statistic and P-value. What conclusion do you get?

Sol'n. *First we exclude the measurements which have difference 0 (measurements 1 and 6). With the remaining $n = 10$ measurements, we can model the signed rank statistic using a normal distribution with mean μ and standard deviation σ:*

$$\mu = \frac{n(n+1)}{4} = \frac{55}{2}, \qquad \sigma = \sqrt{\frac{n(n+1)(2n+1)}{24}} = \sqrt{\frac{385}{4}}.$$

Compute the difference vector Z; rank $|Z|$; and sum the ranks R_i whose corresponding $Z_i > 0$:

$$\begin{aligned} Z &= (-6, 1, 4, -5, 12, 1, 5, -9, 7, 15) \\ R &= (6, 1.5, 3, 4.5, 9, 1.5, 4.5, 8, 7, 10) \\ V &= 1.5 + 3 + 9 + 1.5 + 4.5 + 7 + 10 = 36.5. \end{aligned}$$

Use `R` to compute the one-sided p-value:

```
> pnorm(36.5, mean=55/2, sd=sqrt(385/4), lower.tail=FALSE)
[1] 0.1794757
```

One can also use R's `wilcox.test` function. To get a similar p-value to the normal approximation used above, we must set `correct=FALSE` to disable continuity correction.

```
> X
 [1] 86 71 77 68 91 72 77 91 70 71 88 87
> Y
 [1] 86 77 76 64 96 72 65 90 65 80 81 72
> wilcox.test(X, Y, paired=TRUE, alternative="greater", correct=FALSE)

Wilcoxon signed rank test

data:  X and Y
V = 36.5, p-value = 0.1792
alternative hypothesis: true location shift is greater than 0
```

Parametric vs nonparametric test

6. Comparison of parametric test and nonparametric test.

   (a) Use the following command to generate two vectors `x` and `y`

   ```
   x = rnorm(10,mean = 10, sd =1)
   y = x + rnorm(10,mean=1,sd=1)
   ```

   (b) Perform the paired t-test to test if the difference between x and y (i.e. $x - y$) is less than zero. What is the P-value?

   Sol'n. *The p-value is* 0.0003233.

   ```
   > t.test(x, y, paired=TRUE, alternative="less")

   Paired t-test

   data:  x and y
   t = -5.0983, df = 9, p-value = 0.0003233
   alternative hypothesis: true difference in means is less than 0
   95 percent confidence interval:
          -Inf -0.7314055
   sample estimates:
   mean of the differences
                -1.142020
   ```

   (c) Perform the Wilcoxon signed-rank test to see if the difference between x and y is less than zero. What is the P-value?

   Sol'n. *The p-value is* 0.0009766*, roughly three times the p-value from the t-test.*

   ```
   > wilcox.test(x, y, paired=TRUE, alternative="less")

   Wilcoxon signed rank test

   data:  x and y
   V = 0, p-value = 0.0009766
   alternative hypothesis: true location shift is less than 0
   ```

(d) Repeat the above process 1000 times. How many simulations out of 1000 simulations does the paired t-test give P-value less than 0.05? How about the Wilcoxon signed-rank test? What conclusion can you get from this simulation? Hint: you can use
`t.test(x,y,paired=T,alternative="less")$p.value`
and
`wilcox.test(x,y,paired=T,alternative="less")$p.value`
to get the P-value of the t-test and the Wilcoxon signed-rank test.

Sol'n. *The t-test correctly rejects the null hypothesis more often than the Wilcoxon signed-rank test. This is to be expected since the data are normally distributed.*

```
> n.t <- 0
> n.w <- 0
> for (i in 1:1000) {
+     x <- rnorm(10, mean=10, sd=1)
+     y <- x + rnorm(10, mean=1, sd=1)
+     if (t.test(x, y, paired=TRUE, alternative="less")$p.value < 0.05)
+         n.t <- n.t + 1
+     if (wilcox.test(x, y, paired=TRUE, alternative="less")$p.value < 0.05)
+         n.w <- n.w + 1
+ }
> n.t
[1] 904
> n.w
[1] 868
```

References

[1] Little, M.A., McSharry, P.E., Hunter, E.J., Spielman, J. and Ramig, L.O. (2008). Suitability of dysphonia measurements for telemonitoring of Parkinsons disease. In *IEEE Transactions on Biomedical Engineering*.

[2] Kvam, P.H. and Vidakovic, B. (2007). Nonparametric statistics with applications to science and engineering