Homework #4

Due: Tuesday, October 19 @ 6pm

Please remember to give R code, as well as answers, for any problems where you used R

Problem 1:

We are interested in estimating the concentration, in parts per billion (ppb) of *E. coli* in Lake Michigan on the basis of measurements of a number of samples. Suppose measurements of such samples will be approximately normally distributed with unknown mean (the true concentration) and known SD $\sigma = 1.5$ ppb. How many samples should we take if we wish our 95% CI for the true concentration to have a width ≤ 1 ppb?

Problem 2:

Suppose we measure the log_{10} cytokine response of 15 mice following some treatment, and observe the sample mean $\bar{X} = 1.2$ and sample SD $s = 2.3$

a. Suppose that your null hypothesis is that the population mean $\mu = 0$. Under what circumstances could you use a t-test to test this hypothesis?
b. Assuming the conditions in part (a) hold, what would the t-statistic be?
c. If your *alternative* hypothesis is that the cytokine response is *greater* than 0, what would the p-value be? (Use R.)
d. In words, how would you interpret/describe the result you got in (c)?
e. If your *alternative* hypothesis is that the cytokine response is *different from 0*, what would the p-value be? (Use R.)
f. In words, how would you interpret/describe the result you got in (e)?
g. What is the smallest sample needed for (e) to be significant at the $\alpha = 0.05$ level assuming that everything else (sample SD, sample mean) remains the same? What about for (c)? (You may solve this algebraically *or* by trial and error in R.)

Problem 3:

Review the following functions to describe the confidence interval of the mean and perform a one-sided t-test in R:

```r
# define confidence interval of the mean
mean.conf.int <- function(x, CI = 0.95) {
    xbar <- mean(x)
    n <- length(x) # number of samples
    t.quantile <- qt(1-(1-CI)/2, df = n-1)
    std.error <- sd(x)/sqrt(n)
    conf.int <- c(xbar - t.quantile*std.error, xbar+t.quantile*std.error)
    return(conf.int)
}
```

```r
# Perform a one-sided t-test
# when lower.tail = T, tests if mean of x is significantly LESS than mu
# when lower.tail = F, tests if mean of x is significantly GREATER than mu
# by default, mu = 0 and lower.tail = T
one.sided.t.test <- function(x, mu = 0, lower.tail = TRUE) {
    xbar <- mean(x)
    n <- length(x)
    sampSD <- sd(x)
    tStatistic <- (xbar - mu)/(sampSD/sqrt(n))
    p.value <- pt(tStatistic, df = n - 1, lower.tail = lower.tail)
    return(p.value)
}
```

a. What would happen if you applied these functions to a vector `x` containing NA's?
b. How would you modify these functions to accept data which has NA's in it?
c. Write a function to test if the mean is significantly *different* from μ in *either* direction (greater than or less than). Note: your answer in this part will not count against your grade for this problem, as you will have the opportunity to revise it for part (g) below.
d. Create a vector `testData` (below). Without actually calculating it, would you expect the mean to be significantly different from 0?

```r
testData <- c(-3, rep(-2, 5), rep(-1,10), rep(0,10), rep(1,5),2)
testData
```

```
##  [1] -3 -2 -2 -2 -2 -2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0
## [26]  0  1  1  1  1  1  2
```

e. Using your function from (d), test $H_0 : \mu = 0$ against the alternative that $\mu \neq 0$. What do you get? Is this what you expected? If not, why not?
f. If the answer to (f) was what you expected, great! Rewrite your answer from (d) here for credit. If not, figure out how to modify your answer to part (d) such that it gives you the result you expect in (f), and write that below. Hint: `?abs` may be helpful.
g. Reflect on this problem and write down any observations/lessons learned.

Problem 4:

In problem 3, we wrote our own function to carry out a *t*-test. However, it happens that R already has a function built-in to do that! For this problem, you should first read the help page for the R function `t.test`. At this point, some of its options will be beyond what we've covered, so here we'll focus on the simplest usage.

a. Look at the usage for the "`## Default S3 method`". This is the usage it will default to when you call `t.test` on your data. What arguments *must* be specified (i.e., do not have default values)?
b. For a one-sample test, like we've been doing so far in this homework, we can ignore the `y`, `paired`, and`var.equal` arguments. Suppose you had a sample called `myData`. How would you test:

   - If the mean of `myData` were significantly different from 0?
   - If the mean of `myData` were significantly less than 5?
   - $H_0 : \mu = 0$ vs. $H_A : \mu > 0$

c. Let `myData` be a sample of size 20 drawn from N(3,2). Create this data in R. Using YOUR functions from problem #3 (*not* `t.test`), carry out the tests described in part (b) above.
d. Now, without changing `myData`, carry out these same tests using `t.test` as specified in your answer to (b). Do the *p*-values agree with yours? For the first test, describe each element of its output. What is it telling you?