Reading Data

We will consider daily weather data recorded at the Ann Arbor airport, starting January 1, 2001 and ending December 31, 2006. This data was taken from the Weather Underground website.

1. Begin by downloading "Ann Arbor.txt" from the lab website and save it as a text file in your working directory. You can read this particular dataset into R using the following command:

   ```
   Z=read.table("AnnArbor.txt",sep=",", row.names=NULL, header=T)
   ```

   Each row of Z contains measurements for one day in Ann Arbor.

2. For each 10 degree daily minimum temperature interval from [-20, -11] up to [70,79], calculate the mean and standard deviation of the `Max.Sea.Level.PressureIn`. Note that the daily minimum temperature is found in the 4th column of `Z`, and the `Max.Sea.Level.PressureIn` is found in the 11th column of `Z`. Also, keep track of the number of days (rows in Z) that were used to calculate the mean and variance for each 10 degree interval. Describe your findings.

   ```
   ## Create a sequence of the lower edge of the temp intervals
   range.bottom = seq(from=-20, to=70, by = 10)

   mean.vec = NULL
   var.vec = NULL
   n.obs.vec = NULL

   for ( lower in range.bottom )
   {
     upper = lower + 9

     ## get row indices corresponding to days
     ## with minimum temperatures in the range [lower, upper]
     indices = which( (Z[,4] >= lower) & (Z[,4] <= upper) )

     ## keep track of the number of observations in this interval
     n.obs.vec = c(n.obs.vec, length(indices) )

     ## compute the mean, and variance of the max-seal-level-pressure
     ## then add it to the end of the respective vectors.
     mean.vec = c(mean.vec, mean(Z[indices, 11]) )
     var.vec = c(var.vec, var(Z[indices, 11]))
   }

   ## we now have mean.vec and var.vec,
   ```

```
## we can look at them
mean.vec
var.vec
n.obs.vec
```

Infectious Disease Model

Imagine a simple model for an infectious disease in which each person is classified into one of three categories: "infected," "susceptible," and "dead." Each week, the following processes take place:

- Each infected person dies with a given probability `p.death`.
- Each remaining infected person, infects one randomly-selected susceptible person with probability `p.infect`.
- Each remaining infected person (not infected this week), recovers with probability `p.recover`. (Anyone who recovers is susceptible).

Start with a population of 99 susceptible people and 1 infected person. Write a simulation that iterates over the above steps until nobody is infected. One simulation replication is a model of an "outbreak." Estimate the average number of people that die per outbreak.

Basic tasks we will need:

1. Create a vector `people` corresponding to the 100 people in the population. Let `people[i] = 1` if person i is infected, let `people[i] = 2` if person i is susceptible, and let `people[i] = 3` if person i is dead. Since we start with 99 susceptible people and one infected person, we can initialize `people` as:

   ```
   people = array(2, 100)
   people[1] = 1
   ```

2. Using the `which()` command, find the indices of people that are infected (this indicies vector could be of length 1):

   ```
   infected = which(people == 1)
   ```

3. Of the infected people, figure out who dies and update `people`:

   ```
   p.death = 0.2
   for (j in infected)
   {
     if (runif(1) < p.death) { people[j] = 3 }
   }
   ```

4. Randomly selected one susceptible person to infect

```
susceptible = which(people == 2)
to.be.infected = sample(susceptible)[1]
people[to.be.infected] = 1
```

5. Using these components, put the simulation together:

```
nrep = 1e3
p.recover = 0.1
p.death = 0.2
p.infect = 0.2

num.died = NULL

for (k in 1:nrep)
{
  ## Initialize the population.

  ## 1 = Infected
  ## 2 = Susceptible
  ## 3 = Dead

  people = array(2, 100)
  people[1] = 1

  ## Count the weeks.
  n = 0

  ## Iterate until nobody is infected.
  while (any(people==1))
  {
    n = n+1

    ## Figure out which infected people die
    infected = which(people == 1)
    for (j in infected)
    {
      if (runif(1) < p.death) { people[j] = 3 }
    }

    ## Figure out which remaining infected people infect others,
    ## and if they recover
    infected = which(people == 1)
```

```
    for (j in infected)
    {
      if (runif(1) < p.infect)
      {
        susceptible = which(people == 2)
        if ( sum(people == 2) > 0 )
        {
          to.be.infected = sample(susceptible)[1]
          people[to.be.infected] = 1
        }
      }
      if (runif(1) < p.recover) { people[j] = 2 }
    }
  }

  ## find the number of people that died during this outbreak
  num.died[k] = sum(people == 3)
}
```