Lecture 2 Exercises

Isabel Fulcher

8/3/2018

Part 1: Matrix / Vector Operations

Before starting generate the following in R:

```
set.seed(11)
b <- sample(10,4,replace=TRUE)
A <- matrix(sample(10,16,replace=TRUE),4,4)
```

Now perform the following exercises to get familiar with matrix operations:

1. What is the dimension of $\mathbf{A}$?
2. Find the transpose of $\mathbf{b}$
3. Are vectors in R column or row vectors by default?
4. Calculate the following in R: $\mathbf{A}^T\mathbf{A}$
5. Repeat the previous exercise with a built-in R function
6. Solve $\mathbf{b} = \mathbf{A}\mathbf{x}$ for $\mathbf{x}$
7. Repeat the previous with a built-in R function.
8. Create a $4x4$ matrix with diagonal elements equal to 5 and off-diagonals equal to 0.

Part II: Flow control

For Loops

1. Using a for loop, generate a vector containing the natural numbers up to 10 and add 2 to each element. How can you do this without a loop?
2. Load in the iris dataset using the R code provided below. Use a for loop to compute the standard deviation of the first four columns measuring sepal length, sepal width, pedal length, and pedal width respectively.

```
# load dataset
data(iris)
# view the first ten observations
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Apply functions

1. For question 2 in the "For Loops" section, use `apply` to perform the same task.
2. Report the 20th and 80th percentiles for each variable in the Iris dataset. Hint: use the `quantile` function.
3. Compute the variance-covariance matrix for the four variables in the Iris dataset. Confirm your answer by using the `cov` function. Hint: the covariance matrix is given by $(n-1)^{-1}\mathbf{X}^{*T}\mathbf{X}^{*}$ where $\mathbf{X}^{*}$ contains mean centered values for each of the four variables.
4. In a given library of RNA-seq reads, there are duplicate observations due to PCR amplification. However, you know that there are truly $N = 50,000$ unique reads. Your collaborator wants to know how many reads she should sequence ($K \geq N$) to get a good saturation of her library (i.e. capture as many unique reads as possible). Assume that sampling from the N unique reads occurs with replacement and equal probability. What is the expected number of unique reads for the values of K between 50,000 and 200,000 (use increments of 10,000)?

```
set.seed(345)
```

While Loops and conditional statements

1. Roll a fair six sided die; if the roll is a prime number, print "prime"; if the roll is a composite number, print "composite"; otherwise print "1"

```
#Hint: the %in% function shows if an element is in a vector
2 %in% c(1,2,3)
```

```
## [1] TRUE
```

2. The initial value of the number is 0. If the number is less than 10, then add a random number between 0 and 1 to it and print the resulting sum. Continue until the total sum is greater than 10.

```
#Hint: the runif function returns a random number between 0 and 1
runif(n=1,min=0,max=1)
```

```
## [1] 0.2162537
```

3. The Newton-Raphson method (or Newton Method) is a simple iterative technique for finding the zero r of a real-valued function $f(x)$. First, we provide some initial guess x_0 for r. The goal is then to obtain a better estimate x_n of r in n iterations. Using the Newton-Raphson method, find the 5^{th} root of 7.

- The algorithm can be summarized as follows. First, we can rewrite $r = x_0 + h$ for some h which measures how far the estimate initial guess x_0 is from the true zero r. Assuming h is small, we can use linear approximation to conclude that

  $$0 = f(r) = f(x_0 + h) \approx f(x_0) + hf'(x_0)$$

  It then follows that

  $$h \approx -f(x_0)/f'(x_0)$$

  and therefore a better estimate of r is given by

  $$x_1 = x_0 + h \approx x_0 - f(x_0)/f'(x_0)$$

  Continuing in this way, if x_n is the current estimate of r then x_{n+1} is given by

  $$x_{n+1} \approx x_n - f(x_n)/f'(x_n)$$

- Hint: You may want to specify a small pre-specified tolerance level in place of the approximations. In some cases, it may also be useful to specify a maximum number of iterations.