Lab 0. ZEROTH LAB

Points to Cover

Intro To RStudio

See Video of Lab Zero

Review

Basic Arithmetic

```
# Addition
7 + 2

[1] 9

# Subtraction
7 - 2

[1] 5

# Division
7/2

[1] 3.5

# Multiplication
7 * 2

[1] 14

# Negation
-7

[1] -7

# Exponents
7^2

[1] 49
```

Quotients and Remainders

```
# Remainders
7%%2

[1] 1

# Quotients
7%/%2

[1] 3
```

Trig Functions

```
sin(7)

[1] 0.6569866

cos(7)

[1] 0.7539023

tan(7)

[1] 0.871448
```

Logarithm and Exponential Function

```
# Natural Log
log(7)

[1] 1.94591

# Exponential
exp(7)

[1] 1096.633
```

Lab 1. Numerical Vectors

Review

Vector Assignments

```
c(1, 4, 3.2)  #Combine elements into a vector

[1] 1.0 4.0 3.2

A1 <- c(5, 7, 3, -1)  #Assign a vector to variable A1
# Variable names can only contain letters, numbers, underscores, and periods, are case-sensitive, and must start with
# a letter or a period followed by a letter.
A1

[1]  5  7  3 -1
```

Sequential Vectors

```
4:9  #increments of 1

[1] 4 5 6 7 8 9

seq(from = 4, to = 6.9, by = 0.5)  # increments of .5 starting at 4. Ends at 6.5.

[1] 4.0 4.5 5.0 5.5 6.0 6.5

seq(from = 4, to = 6, length = 9)  # 9 equally spaced elements starting at 4 and ending at 6.9

[1] 4.00 4.25 4.50 4.75 5.00 5.25 5.50 5.75 6.00
```

Replicating Vectors

```
rep(c(-4.9, -7, -4.5), times = 256)  # replicate the vector c(-4.9,-7.0,-4.5) 256 times
```

Combine Vectors

```
x <- 7:4  #sequence vector
y <- rep(9, times = 3)  #replicated vector
c(x, y, 10, x)  #combining x,y, and 10

 [1]  7  6  5  4  9  9  9 10  7  6  5  4
```

Filtering Elements

```
x <- 11:20
x

 [1] 11 12 13 14 15 16 17 18 19 20

x[c(9, 1)]  #filter the 9th and then the 1st element

[1] 19 11

x[-c(9, 1)]  #Filter everything except the 9th and 1st element

[1] 12 13 14 15 16 17 18 20
```

Vector Length

```
x <- c(12, 122, 122, 14, 15, 15, 15, 15, 12, 122)
length(x)  #number of elements in x

[1] 10
```

Unique elements

```
x <- c(12, 122, 122, 14, 15, 15, 15, 15, 12, 122)
unique(x)  #unique elements in x

[1]  12 122  14  15
```

Table elements

```
x <- c(12, 122, 122, 14, 15, 15, 15, 15, 12, 122)
table(x)  #frequencies of/count elements in x

x
 12  14  15 122
  2   1   4   3
```

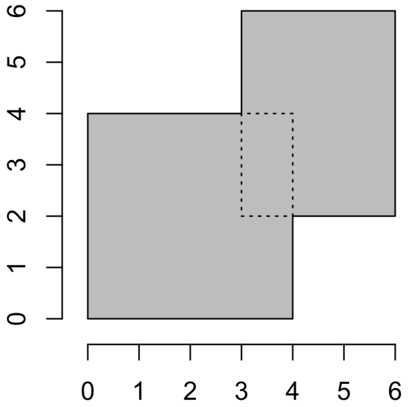
Vectorized Operations & Recycling

```
x <- c(2, 3)
y <- c(1, 2, 3, 4)
y^x  # c(1^2 , 2^3 , 3^2 , 4^3)

[1]  1  8  9 64

log(y)  #c(log(1), log(2), log(3), log(4))

[1] 0.0000000 0.6931472 1.0986123 1.3862944
```

Examples To Work Through During Lab

1. Suppose $P(A) = .8, P(B) = .6$ and $P(A \cap B) = .55$. Determine $P(A$ or $B)$.

2. Compute the area of the shaded region below. (All angles are right angles.)



3. In the following table, $\sum prob = 1$. Use the table to find the number mu,

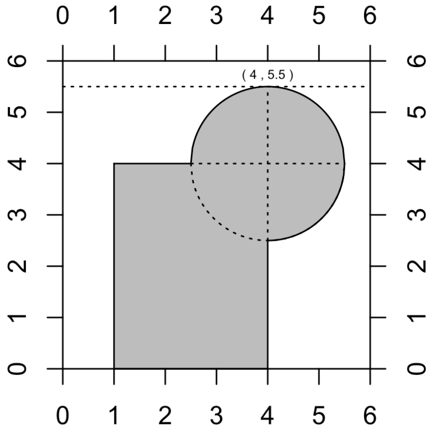
$$\mu = \sum(x \times prob).$$

x	prob
4	0.391
3	0.288
2	0.180
1	0.141

Problems For You to Work on

4. Suppose $P(A \text{ or } B) = .85$, $P(B) = .7$ and $P(A \cap B) = .35$. Determine $P(A)$.

5. Compute the area of the shaded region below. (All angles are right angles. All circular things are circles. Pi is pi in r-code.)



6. Using the given table, compute the following:

$$\sigma = \sqrt{\sum((x-\mu)^2 \times prob)}$$

x	prob
4	0.391
3	0.288
2	0.180
1	0.141

7. Using at most two lines of code, compute the square of the natural log of the first 1000 integers. The natural log is computed with the $log(\ \ )$ function. It is vectorized.

$$\mathtt{ln}(1)^2,\mathtt{ln}(2)^2,\mathtt{ln}(3)^2,\dots,\mathtt{ln}(1000)^2,$$