

Lab 1 = 1/1; Lab 2 = 1.5/1.5; Lab 3 = 1.5/1.5; Total = 4/4, nice paper

BIOS 6621 HW 01 – R introduction

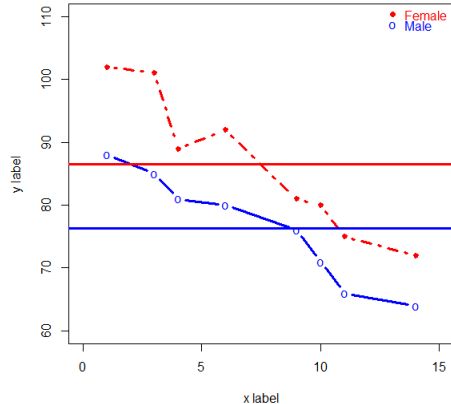
Turn in annotated R code and output for the exercises below from the R labs 1-3 posted in the Rstuff folder. Please submit a Word or PDF file to HW 01 on CANVAS.

Lab 1: Ex 1, 2

```
> ### BIOS6621 Homework1 Question1
> ### Basic graphing
> x <- c(1,3,4,6,9,10,11,14)
> y1 <- c(102,101,89,92,81,80,75,72)
> y2 <- c(88,85,81,80,76,71,66,64)
> plot( x, y1, xlab = 'x label', ylab = 'y label', pch = 19,
+       type = "b", cex = 0.7, col = "red", lty = 4, lwd = 3,
+       xlim = c(0, 15), ylim = c(60, 110) )
> points( x, y2, pch = "o", cex = 1, col = "blue", lty = 1,
+         type = "b", lwd = 3 )
> ### Ex 1: Add a red horizontal line at mean(y1) and a blue horizontal line at mean(y2)
> abline( h = mean(y1), cex = 0.7, lty = 1, col = "red", lwd = 3)
> abline( h = mean(y2), cex = 0.7, lty = 1, col = "blue", lwd = 3)
> ### Ex 2: Move the legend up on the graph so it looks better and add red and blue points in front of the text
> legend( "topright", leg = c("Female", "Male"),
+        text.col = c("red", "blue"), col = c("red", "blue"),
+        bty = "n", pch = c(19, 11), cex = c(1, 1) )
```

Right

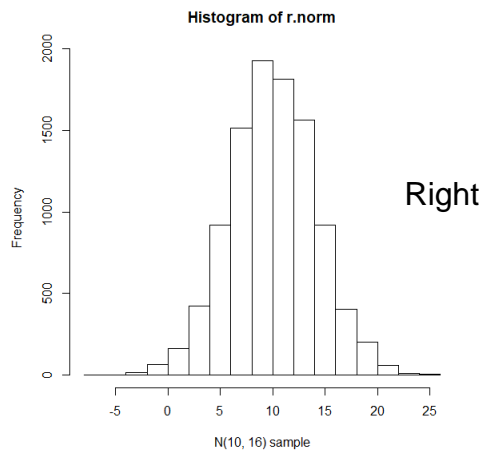
Right



Lab 2: Ex 1, 2, 3

```
> # Ex 1: Use rep and seq to create c(.2,.2,.4,.4,.6,.6,.2,.2,.4,.4,.6,.6,.2,.2,.4,.4,.6,.6)
> rep( rep( seq( 0.2, 0.6, by = 0.2), each = 2), times = 3)
[1] 0.2 0.2 0.4 0.4 0.6 0.6 0.2 0.2 0.4 0.4 0.6 0.6 0.2 0.2 0.4 0.4 0.6 0.6
> # Ex 2: Generate a sample of 10000 normal values with mean 10 and sd 4
> # Make a histogram of the values
> # Check that the empirical mean and SD match those used to generate the sample
> # For  $X \sim N(10, 16)$ , find  $\Pr(X > 18)$ 
> # For  $X \sim N(10, 16)$ , find the value so that 97.5% of the distribution is less than that value
> # Make a smooth line graph of the  $N(10, 16)$  density. (Hint: Use seq)
> set.seed( seed = 5 )
> r.norm <- rnorm( n = 10000, mean = 10, sd = 4 )
> hist( r.norm, xlab = "N(10, 16) sample" )
```

Right



Right

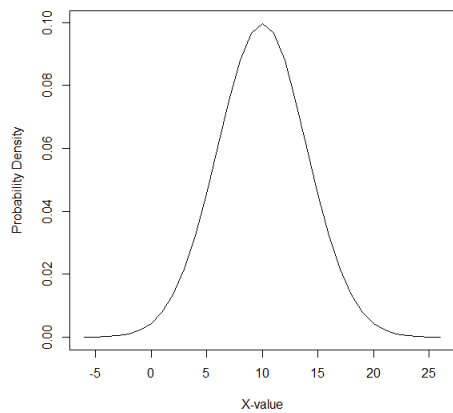
```
> mean( r.norm ); sd( r.norm );
[1] 10.00724
[1] 4.048991
> p.norm <- pnorm( q = 18, mean = 10, sd = 16 ); p.norm
[1] 0.6914625
> q.norm <- qnorm( p = .975, mean = 10, sd = 4 ); q.norm
[1] 17.83986
> x <- seq( 10 - 4 * 4, 10 + 4 * 4, by = 1 )
> d.norm <- dnorm( x, mean = 10, sd = 4 );
> plot( x, d.norm, xlab = "X-value", ylab = "Probability Density", type = "l" )
```

Right

Typo, sd=4

Right

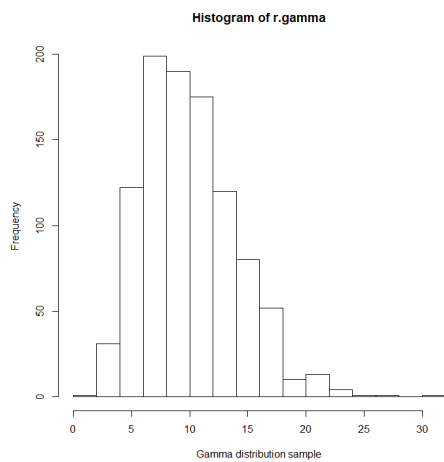
Right



Right

```
> # Ex 3: Generate a sample of 10000 values from a Gamma distribution with mean 10 and sd 4
> #   Verify empirically that your sample mean and sd are close to 10 and 4
> #   This is very useful, when working with a new distn in R (or SAS or ...) it's good to do this
> set.seed( seed = 55 )
> mu = 10; sigma = 4
> s <- sigma ^ 2 / mu; a <- mu / s;
> r.gamma <- rgamma( n = 1000, shape = a , scale = s)
> hist( r.gamma, xlab = "Gamma distribution sample")
> mean( r.gamma ); sd( r.gamma );
[1] 10.01178
[1] 4.013219
```

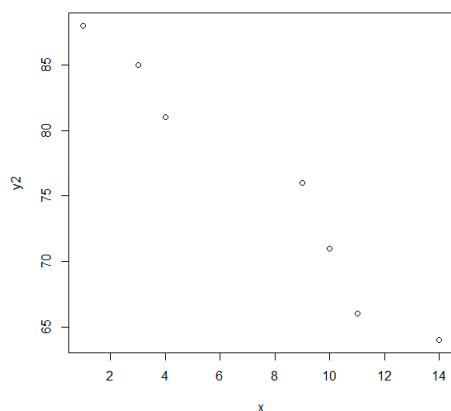
Right



Lab 3: Ex 1, 2, 3

```
> # Ex 1: Explain (1 sentence each) what each of these statements does
> x[c(3:7)-2]
[1] 1 3 4 6 9
> # to show the values of subset: the element 1 to element 5. Right
> x[c(3:7)]-2
[1] 2 4 7 8 9
> # to show the values of subset: the element 3 to element 7 minus 2. Right
> xy.mat[14, ]
> # to show the row 14, the whole columns Right, except no row 14 so gives an error
> xy.mat[c(2:4)]
[1] 3 4 6
> # to show the row 2, 3, 4, column 1 values ( x values)
> cbind(x, lab)
> # to show the column x and column lab together. Right but better to specify row and c
```

```
> # Ex 2: Using the object xy.mat and not making any new assignments (don't use <-),
> # graph y2 versus x, omitting the 4th row, and with axis labels "y2" and "x"
> x; y
[1] 1 3 4 6 9 10 11 14
[1] 100 101 102 103 104 105 106 107 108 109 110 111
> plot(y2[-4] ~ x[-4], ylab = "y2", xlab = "x" )
```



Right

```
> # Ex 3: Suppose y1 and y2 are measures of an outcome at times 1 and 2 for 8 subjects.
> # So cbind(y1,y2) is the 'wide' form of the dataset
> # Create the 'long' form of the dataset, with 16 rows and 3 columns,
> # a column for y, one for time, and one for subject id.
> y.long <- rbind( c( y1, y2 ) ); y.long
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16]
[1,] 102 101 89 92 81 80 75 72 88 85 81 80 76 71 66 64
> rownames( y.long ) <- c("y-value"); y.long
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16]
y-value 102 101 89 92 81 80 75 72 88 85 81 80 76 71 66 64
> y.time <- rep( c( 1, 2 ), each = 8 ); y.time
[1] 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
> sub.ID <- rep( 1:8, time=2 ) ; sub.ID
[1] 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
> y.mat <- rbind( sub.ID, y.time, y.long ); y.mat
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16]
sub.ID 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
y.time 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
y-value 102 101 89 92 81 80 75 72 88 85 81 80 76 71 66 64
> y.mat.t <- t( y.mat ); y.mat.t
```

	sub.ID	y.time	y-value
[1,]	1	1	102
[2,]	2	1	101
[3,]	3	1	89
[4,]	4	1	92
[5,]	5	1	81
[6,]	6	1	80
[7,]	7	1	75
[8,]	8	1	72
[9,]	1	2	88
[10,]	2	2	85
[11,]	3	2	81
[12,]	4	2	80
[13,]	5	2	76
[14,]	6	2	71
[15,]	7	2	66
[16,]	8	2	64

Right