Consulting Homework 1

Tim Vigers

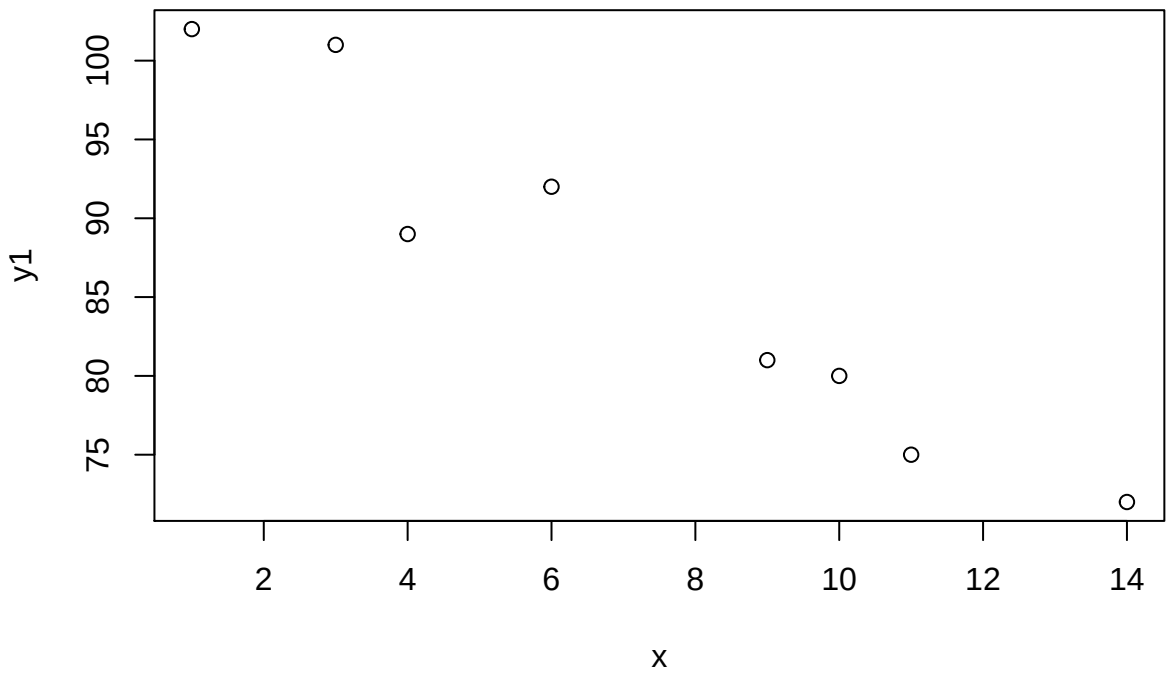
September 2, 2018

R Lab 1

Ex. 1 & 2

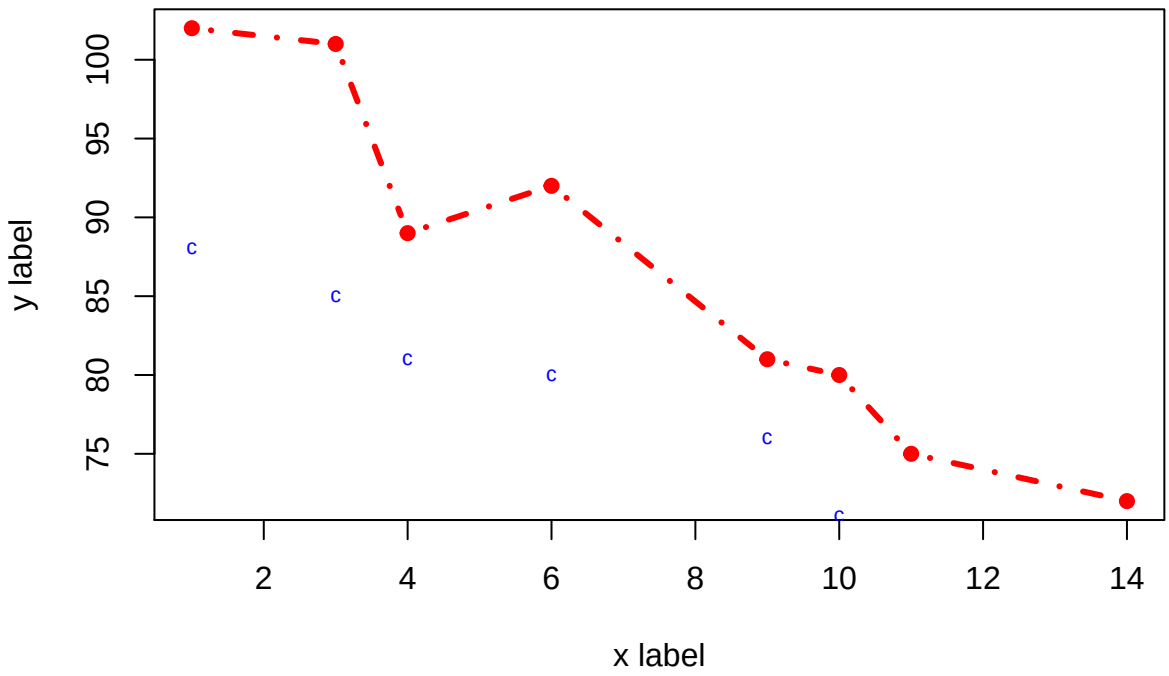
Add a red horizontal line at mean(y1) and a blue horizontal line at mean(y2). Move the legend up on the graph so it looks better and add red and blue points in front of the text.

```
# Code below copied from R Lab 1 document.
# Create 3 vectors of length 8
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)
### Basic x-y graph
plot(x, y1)
```

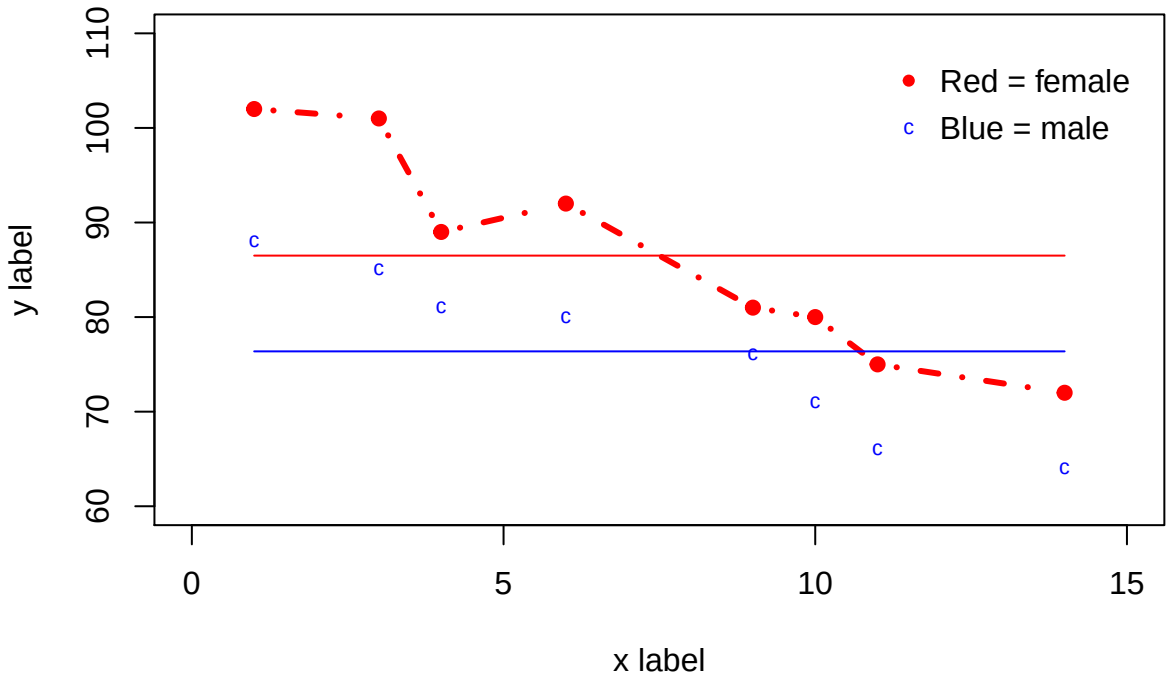



```
### Useful options to plot
### Documentation in Help menu
### ?par gives options
plot(x, y1, xlab='x label', ylab='y label', pch=19, type="b", cex=.7, col="red", lty=4, lwd=3)
```

```
### Adding things to a plot
points(x, y2, pch="circle", cex=.7, col="blue")
```




```
### Note that some points are outside the plot.  To fix, set axis limits
plot(x, y1, xlab='x label', ylab='y label', pch=19, type="b", cex=.7, col="red",
      lty=4, lwd=3, xlim=c(0, 15), ylim=c(60, 110))
points(x, y2, pch="circle", cex=.7, col="blue")
# Add a red line at mean(y1).
lines(x,rep(mean(y1),8),col = "red")
# Add a blue line at mean(y2).
lines(x,rep(mean(y2),8),col = "blue")
# Move the legend (deleted previous legend).
text(12, 105, "Red = female", adj=0)
text(12, 100, "Blue = male", adj=0)
# Add red point in front of the text.
points(11.5, 105, pch=19, type="b", cex=.7, col="red")
# Add blue point in front of the text.
points(11.5, 100, pch="circle", cex=.7, col="blue")
```


y label
110
100
90
80
70
60
0
5
10
15
x label
Red = female
Blue = male
c

R Lab 2

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)

```
# Make the sequence we're aiming for.
goal <- c(.2,.2,.4,.4,.6,.6,.2,.2,.4,.4,.6,.6,.2,.2,.4,.4,.6,.6)
# Use seq() to create c(0.2, 0.4, 0.6).
seq <- seq(.2,.6,by = 0.2)
# Use rep() to repeat each element in seq twice.
seq <- rep(seq,each = 2)
# Use rep() again to repeat the whole list three times.
seq <- rep(seq,times = 3)
# Show the sequence
seq
```

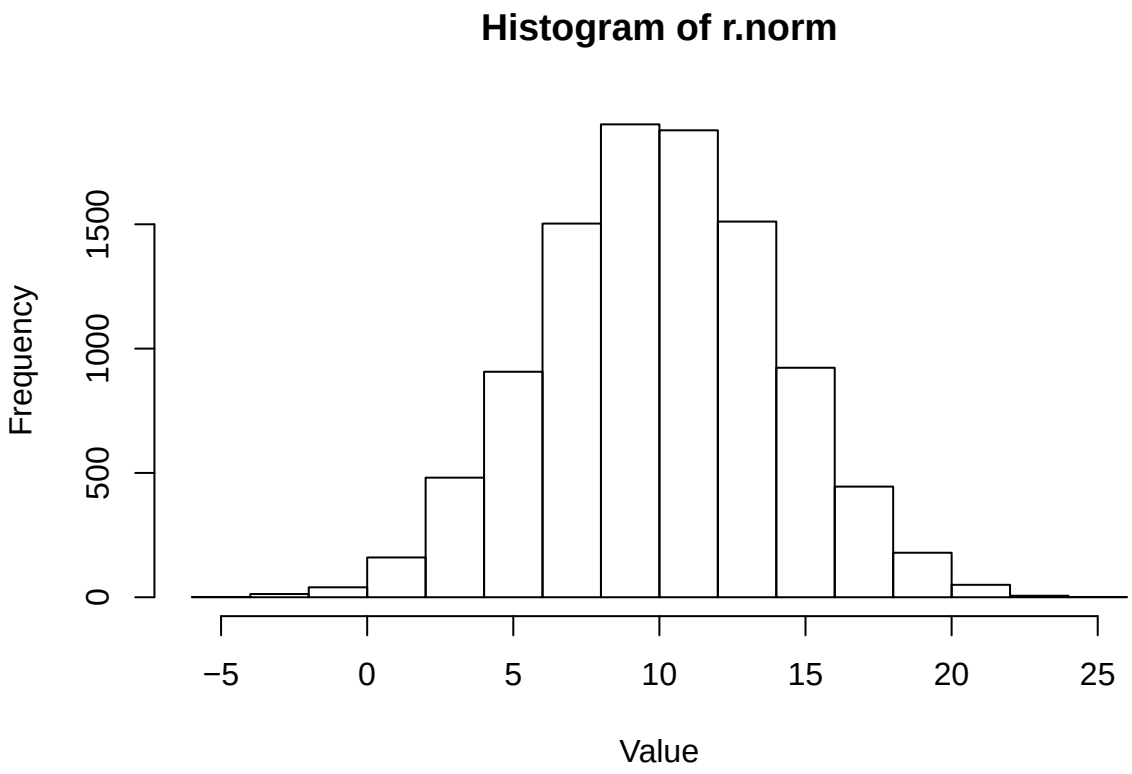
```
##  [1] 0.2 0.2 0.4 0.4 0.6 0.6 0.2 0.2 0.4 0.4 0.6 0.6 0.2 0.2 0.4 0.4 0.6
## [18] 0.6
```

```
# Double check that it matches the goal sequence.
seq == goal
```

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

Ex. 2

```
# Generate a sample of 10000 normal values with mean 10 and sd 4.
r.norm <- rnorm(n = 10000, mean = 10, sd = 4)
# Make a histogram of the values.
hist(r.norm,xlab = "Value")
```




```
# Check that the empirical mean and SD match those used to generate the sample.
mean(r.norm)
```

```
## [1] 10.00698
```

```
sd(r.norm)
```

```
## [1] 4.004099
```

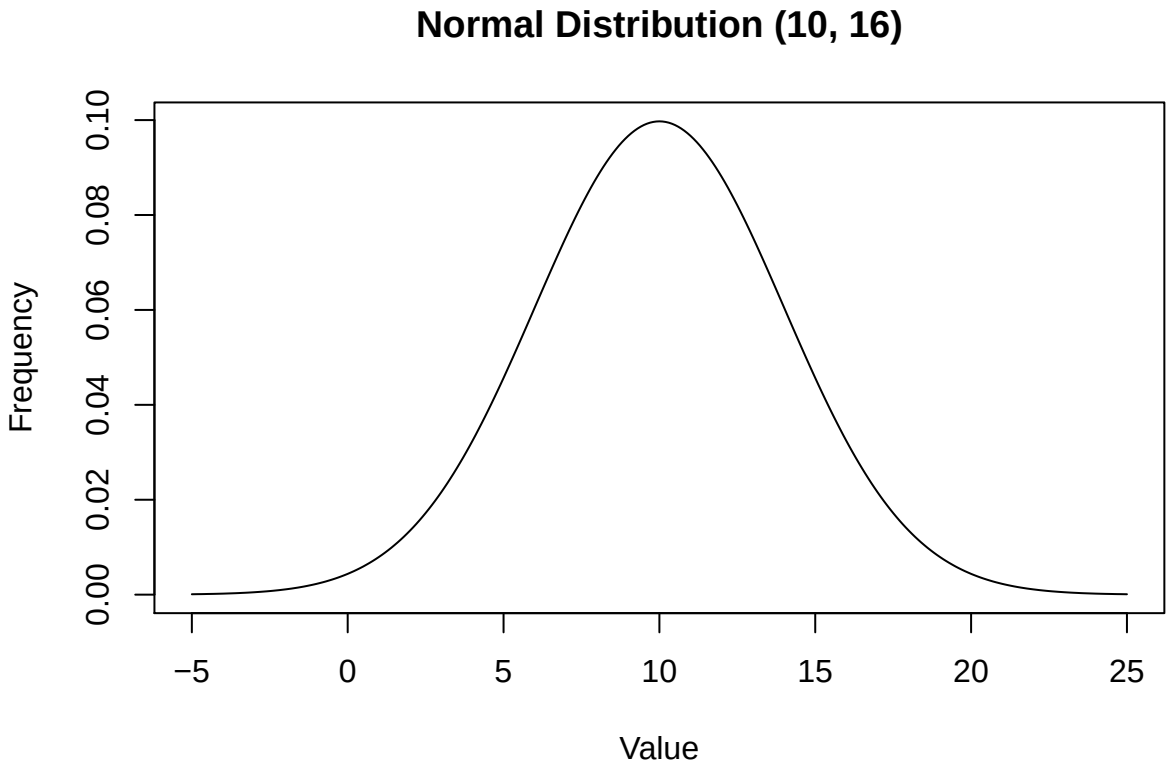
```
# For X~N(10, 16), find Pr(X > 18).
pnorm(18,mean = 10,sd = 4,lower.tail = FALSE)
```

```
## [1] 0.02275013
```

```
# For X~N(10, 16), find the value so that 97.5% of the distribution is less than that value.
qnorm(0.975,mean = 10,sd = 4)
```

```
## [1] 17.83986
```

```
# Make a smooth line graph of the N(10, 16) density.
x <- seq(-5,25, by = 0.1)
y <- dnorm(x, mean = 10, sd = 4)
plot(x, y, type="l", main = "Normal Distribution (10, 16)", ylab = "Frequency", xlab = "Value")
```



Ex. 3

With a shape parameter k and a scale parameter s, for a Gamma distribution:

$$\mu = ks = 10 \text{ and } \sigma^2 = ks^2 = 16$$

So:

$$k = \frac{100}{16} \text{ and } s = \frac{16}{10}$$

```
# Generate a sample of 10000 values from a Gamma distribution with mean 10 and sd 4.
r.gamma <- rgamma(n = 10000, shape = 25/4, scale = 8/5)
# Verify empirically that your sample mean and sd are close to 10 and 4.
mean(r.gamma)
```

```
## [1] 10.02062
```

```
sd(r.gamma)
```

```
## [1] 3.985463
```

R Lab 3

Ex. 1

Explain (1 sentence each) what each of these statements does:

a. x[c(3:7)-2]: This command selects elements (3-2), (4-2), (5-2), (6-2), and (7-2), (i.e. elements 1 - 5) from the vector x.

```
x
```

```
## [1]  1  3  4  6  9 10 11 14
```

```
x[c(3:7)-2]
```

```
## [1] 1 3 4 6 9
```

b. x[c(3:7)]-2: This takes elements 3 - 7 of vector x, and subtracts 2 from each value.

```
x[c(3:7)]
```

```
## [1]  4  6  9 10 11
```

```
x[c(3:7)]-2
```

```
## [1] 2 4 7 8 9
```

c. xy.mat[14,]: This takes row 14 and all columns of the matrix xy.mat (subsetting row 8 is shown below, since there are only 8 rows of our matrix).

```
xy.mat[8, ]
```

```
##  x y1 y2
## 14 72 64
```

d. xy.mat[c(2:4)]: This takes the values in column 1, rows 2, 3, and 4.

```
xy.mat[c(2:4)]
```

```
## [1] 3 4 6
```

e. cbind(x, lab): This takes the vectors x and lab, and turns them into a matrix where each vector is a column.

```
cbind(x, lab)
```

```
##      x    lab
## [1,] "1"  "a"
## [2,] "3"  "b"
## [3,] "4"  "c"
## [4,] "6"  "d"
## [5,] "9"  "e"
## [6,] "10" "f"
## [7,] "11" "g"
## [8,] "14" "h"
```

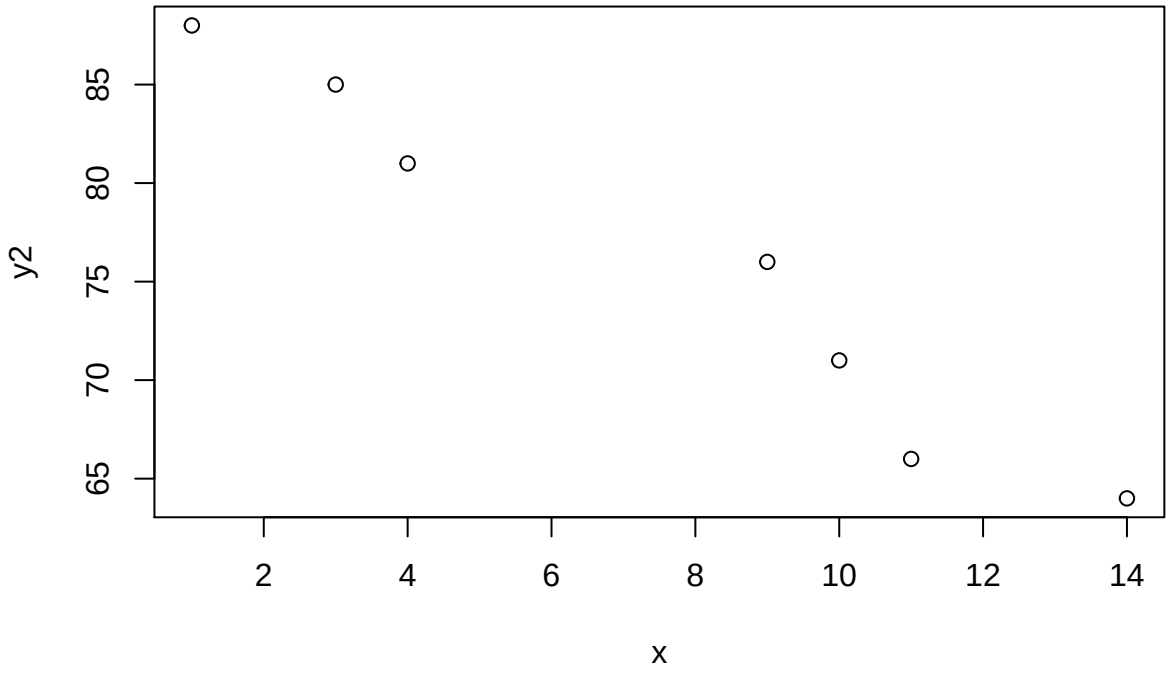
Ex. 2

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".

```
xy.mat
```

```
##       x  y1 y2
## [1,]  1 102 88
## [2,]  3 101 85
## [3,]  4  89 81
## [4,]  6  92 80
## [5,]  9  81 76
## [6,] 10  80 71
## [7,] 11  75 66
## [8,] 14  72 64
```

```
plot(xy.mat[c(1:3,5:8),1], xy.mat[c(1:3,5:8),3], xlab = "x", ylab = "y2")
```



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.

```
# Make an empty matrix.
mat <- matrix(NA, ncol=3, nrow=16)
# Name the columns.
colnames(mat) <- c("subjectid", "time", "outcome")
# Fill in subject ids (from vector x).
mat[,1] <- rep(x,times = 2)
```

```r
# Fill in times 1 and 2.
mat[,2] <- c(rep(1,times = 8),rep(2,times = 8))
# Fill in outcomes data from y1 and y2.
mat[,3] <- c(y1,y2)
# Show the matrix.
print.data.frame(as.data.frame(mat),row.names = FALSE)
```

```
##  subjectid time outcome
##          1    1     102
##          3    1     101
##          4    1      89
##          6    1      92
##          9    1      81
##         10    1      80
##         11    1      75
##         14    1      72
##          1    2      88
##          3    2      85
##          4    2      81
##          6    2      80
##          9    2      76
##         10    2      71
##         11    2      66
##         14    2      64
```