R Warmup

Henry Frye

Objects and Arithmetic

Exercise 1

Part a

This will result in the length of vector x

```
x<-c(4,2,6)
y<-c(1,0,-1)
length(x)
```

```
## [1] 3
```

Part b

This will result in the sum of vector x.

```
sum(x)
```

```
## [1] 12
```

Part c

This will result in the sum of the square for each value within vector x.

```
sum(x^2)
```

```
## [1] 56
```

Part d

This will result in the sum of each value between vectors x and y based on position within the vectors

```
x+y
```

```
## [1] 5 2 5
```

Part e

This will result in the product of each value between vectors x and y based on position within the vectors

```
x*y
```

```
## [1]  4  0 -6
```

Part f

This will subtract 2 from each value in vector x.

```
x-2
```

```
## [1] 2 0 4
```

Part g

This will square each value in vector x.

```
x^2
```

```
## [1] 16  4 36
```

Exercise 2

Part a

This will give a sequence of whole integers [7,11].

```
7:11
```

```
## [1]  7  8  9 10 11
```

Part b

This will give a sequence of whole integers [2,9].

```
seq(2,9)
```

```
## [1] 2 3 4 5 6 7 8 9
```

Part c

This will give a sequence of integers counting by 2 from 4 to 10.

```
seq(4,10,by=2)
```

```
## [1]  4  6  8 10
```

Part d

This will give a sequence of integers between and 3 and 30 divided into 10 parts.

```
seq(3,30,length=10)
```

```
##  [1]  3  6  9 12 15 18 21 24 27 30
```

Part e

This will give a descending sequence from 6 to -4 counting from 2's.

```
seq(6,-4,by=-2)
```

```
## [1]  6  4  2  0 -2 -4
```

Summaries and Subscripting

Exercise 1

Part a

This will return the second value of vector x.

```
x<- c(5,9,2,3,4,6,7,0,8,12,2,9)
x[2]
```

```
## [1] 9
```

Part b

This will return the second through fourth values of vector x

```
x[2:4]
```

```
## [1] 9 2 3
```

Part c

This will return the second, the third, and the sixth value of vector x.

```
x[c(2,3,6)]
```

```
## [1] 9 2 6
```

Matrices

Exercise 1

Setup

```
x <- matrix(c(3,2,-1,1), nrow = 2)
print(x)
```

```
##      [,1] [,2]
## [1,]    3   -1
## [2,]    2    1
```

```
y<- matrix(c(1,4,0,0,1,-1), nrow = 2)
print(y)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    1
## [2,]    4    0   -1
```

Part a

This will multiply each value in matrix **x** by 2.

```
2*x
```

```
##      [,1] [,2]
## [1,]    6   -2
## [2,]    4    2
```

Part b

This will result in the square of **x**.

```
x*x
```

```
##      [,1] [,2]
## [1,]    9    1
## [2,]    4    1
```

Part e

This will trnanspose matrix **y**.

```
t(y)
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    0    0
## [3,]    1   -1
```

Exercise 2

Part a

```
x[1,]
```

```
## [1]  3 -1
```

Part b

```
x[2,]
```

```
## [1] 2 1
```

Part c

```
x[,2]
```

```
## [1] -1  1
```

Part d

```
y[1,2]
```

```
## [1] 0
```

Part e

```
y[,2:3]
```

```
##      [,1] [,2]
## [1,]    0    1
## [2,]    0   -1
```

Statistical Computation and Simulation

Exercise 1

Part a

```
dnorm(x = .5, mean = 2, sd = .25)
```

```
## [1] 2.430353e-08
```

Part b

```
pnorm(2.5, mean = 2, sd = .25)
```

```
## [1] 0.9772499
```

Part c

```
qnorm(.95, mean =2, sd = .25)
```

```
## [1] 2.411213
```

Part d

```
pnorm(3, mean = 2, sd = .25) - pnorm(1, mean = 2, sd = .25)
```

```
## [1] 0.9999367
```

Exercise 3

```
pois.sim <- rpois(n = 100, lambda = 5)
summary(pois.sim)
```

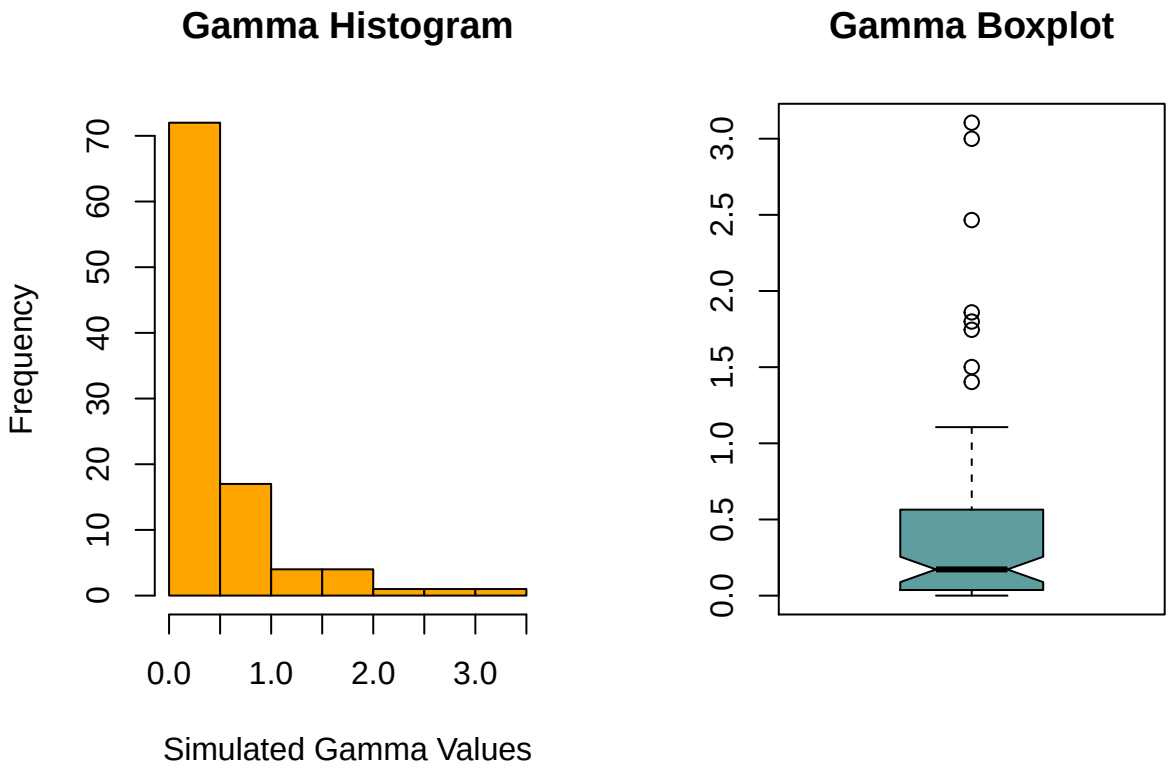
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    3.75    5.00    5.08    7.00   10.00
```

Graphics

Exercise 1

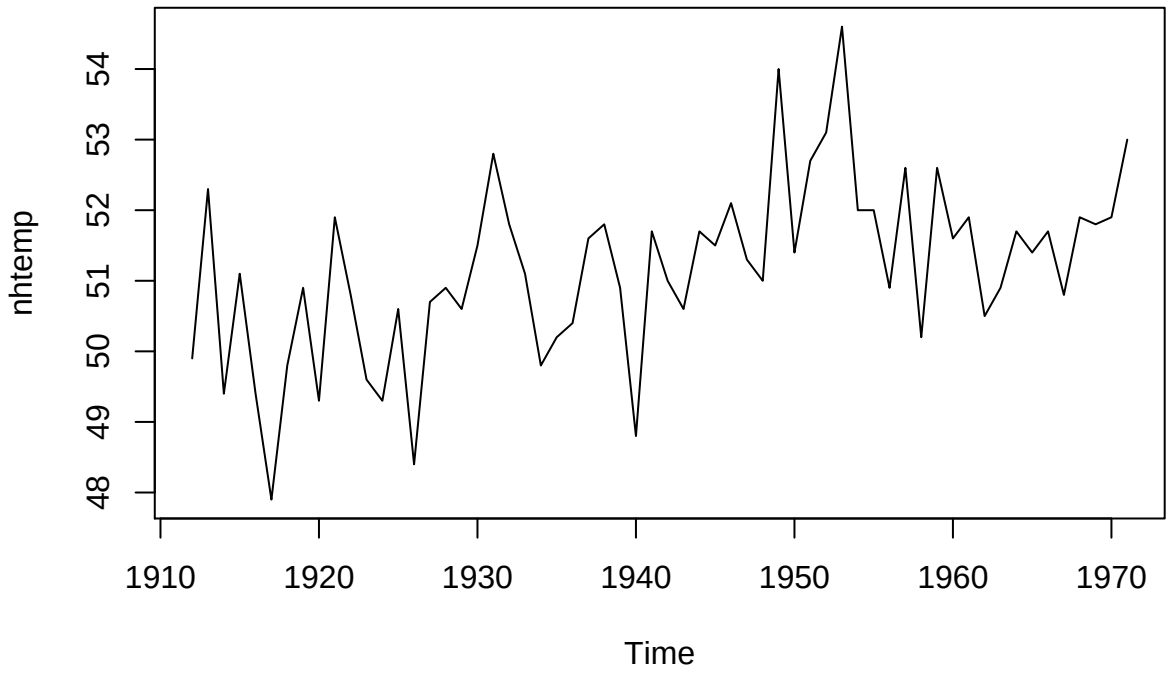
```
gam.sim <- rgamma(n = 100, shape = .5)

par(mfrow = c(1,2))
hist(gam.sim, xlab = 'Simulated Gamma Values', col = 'orange',
     main = 'Gamma Histogram')
boxplot(gam.sim, notch = TRUE, col = 'cadetblue', main = 'Gamma Boxplot', main.cex = .5)
```



Exercise 3

```
data(nhtemp)
plot(nhtemp)
```


nhtemp
48
49
50
51
52
53
54
1910
1920
1930
1940
1950
1960
1970
Time