MATH 3070 Lab Project 3

Aini Liang

September 1, 2016

Remember: I expect to see commentary either in the text, in the code with comments created using `#`*, or (preferably) both!* ***Failing to do so may result in lost points!***

Problem 1

The `faithful` *(Package* ***datasets*** *is built in Base R) dataset records the waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park. Find the mean waiting time, median waiting time, longest waiting time and shortest waiting time in minutes.*

```
# Your code here
data(faithful)
attach(faithful)
mean(waiting)
```

```
## [1] 70.89706
```

```
median(waiting)
```

```
## [1] 76
```

```
range(waiting)
```

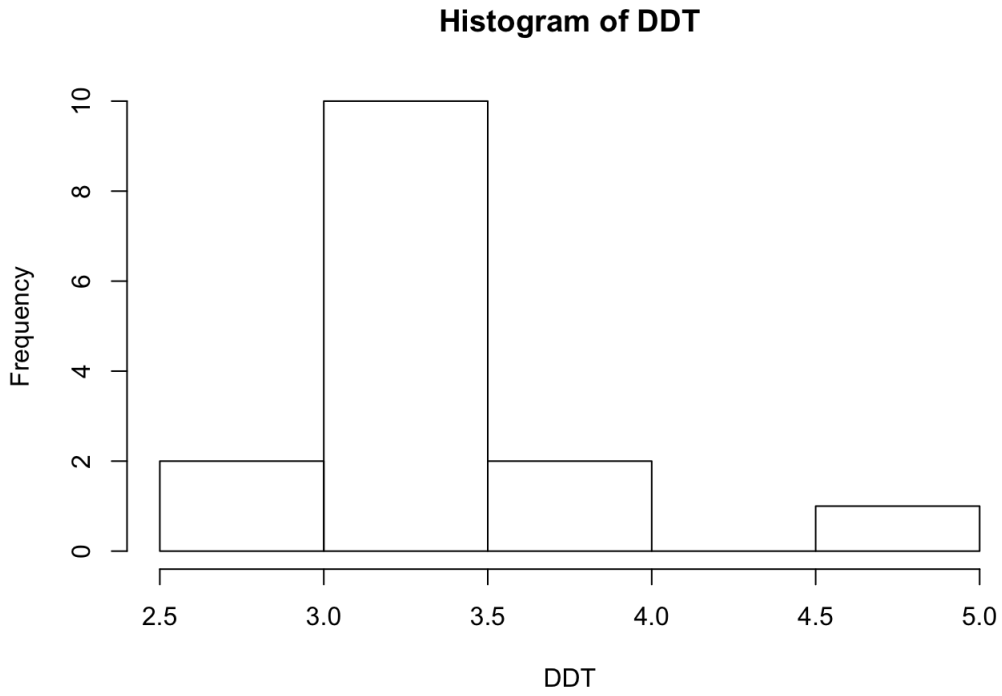
```
## [1] 43 96
```

```
detach(faithful)
#use mean(),median(),range()to find the mean waiting time, median waiting time, longest waiting time and sho
rtest waiting time in minutes
```

Problem 2 (Verzani problem 2.34)

The data set `DDT` *(****MASS****) contains independent measurements of the pesticide DDT on kale. Make a histogram and a boxplot of the data. From these, estimate the mean and standard deviation. Check your answers with the appropriate functions.*

```
# Your code here
#install.packages("MASS")
library(MASS)
#View(DDT)
hist(DDT)
```



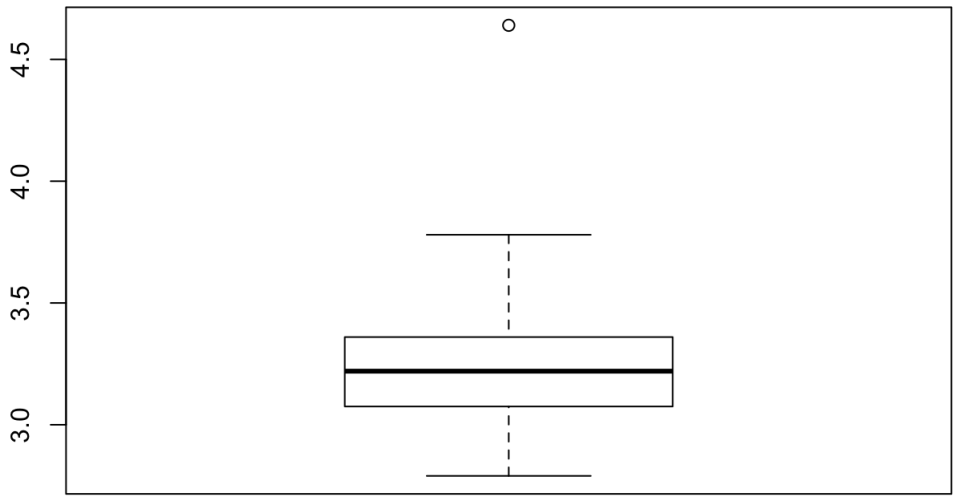

```
mean(DDT)
```

```
## [1] 3.328
```

```
sd(DDT)
```

```
## [1] 0.4371531
```

```
boxplot(DDT)
```




```
#use hist() and boxplot() to make a histogram and a boxplot of the data
```

Problem 3 (Verzani problem 2.32 modified)

*Write a function 'density_compare()' that fits a density estimate to a given data set and plots that estimate along with the appropriate histogram of the given data set. Try your function with the data set `pi2000` (**UsingR**). Why might you want to add an argument like `breaks = 0:10-.5` to `hist()`? (Hint: read the documentation of `hist()` to see what setting this argument does). Feel free to add other parameters to your plot methods to see how they can be changed.*

```
# Your code here  break make
#install.packages(UsingR)
library(UsingR)
```

```
## Loading required package: HistData
```

```
## Loading required package: Hmisc
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
##
## Attaching package: 'UsingR'
```

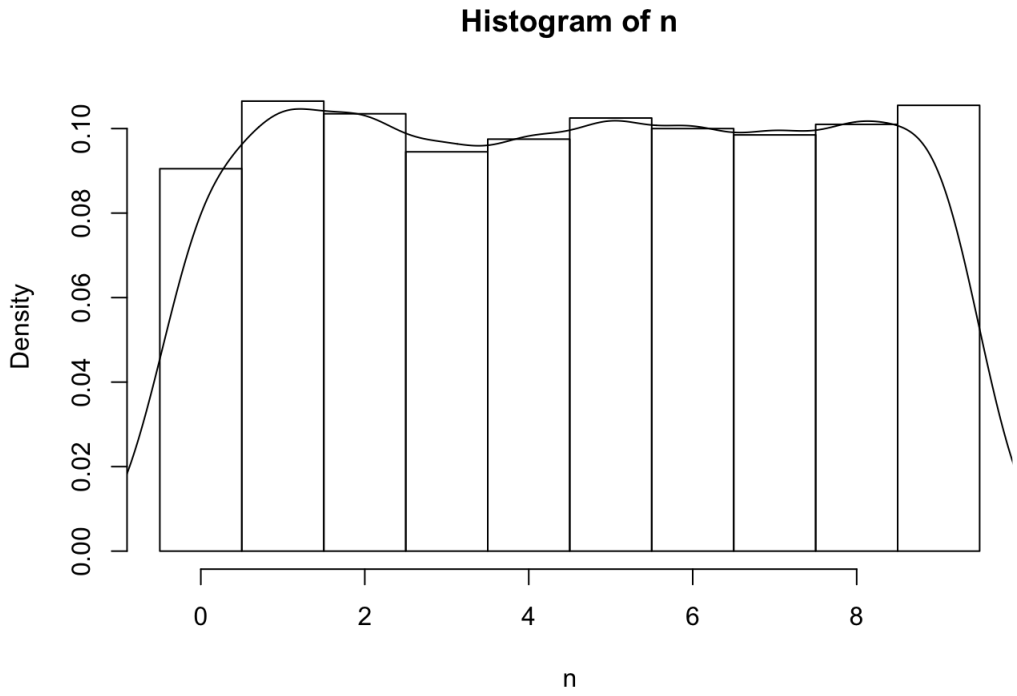
```
## The following object is masked from 'package:survival':
##
##     cancer
```

```
#View(pi2000)

density_compare=function(n){


  hist(n,freq = FALSE,breaks = 0:10-.5)
lines(density(n))
}

density_compare(n=pi2000)
```




```
#make more beautiful
```

BONUS (Verzani problem 2.25)

Write a function `isprime()` *that checks if a number \(x\) is prime by dividing \(x\) by all the values in \(2, ..., x - 1\) then checking to see if there is a remainder of 0. The expression* `a %% b` *returns the remainder of* `a` *divided by* `b`*.*

```
# Your code here
```

You do not need to check all numbers from 2 to \(x - 1\) to see if a number is prime. What is the largest you would need to go for an arbitrary \(x\)? Create a new function, `isprime2()`*, that implements this better (yet still slow) method.*

```
# Your code here
```