Code

Castellano_CS636_Lab04

1. **IRIS**

(1) How many features are numerical data? And how many are categorical data

Hide

```
str(iris[0,])
```

```
'data.frame':	0 obs. of  5 variables:
 $ Sepal.Length: num 
 $ Sepal.Width : num 
 $ Petal.Length: num 
 $ Petal.Width : num 
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 
```

There are 5 features, 4 numerical and 1 categorical.

(2) Histogram

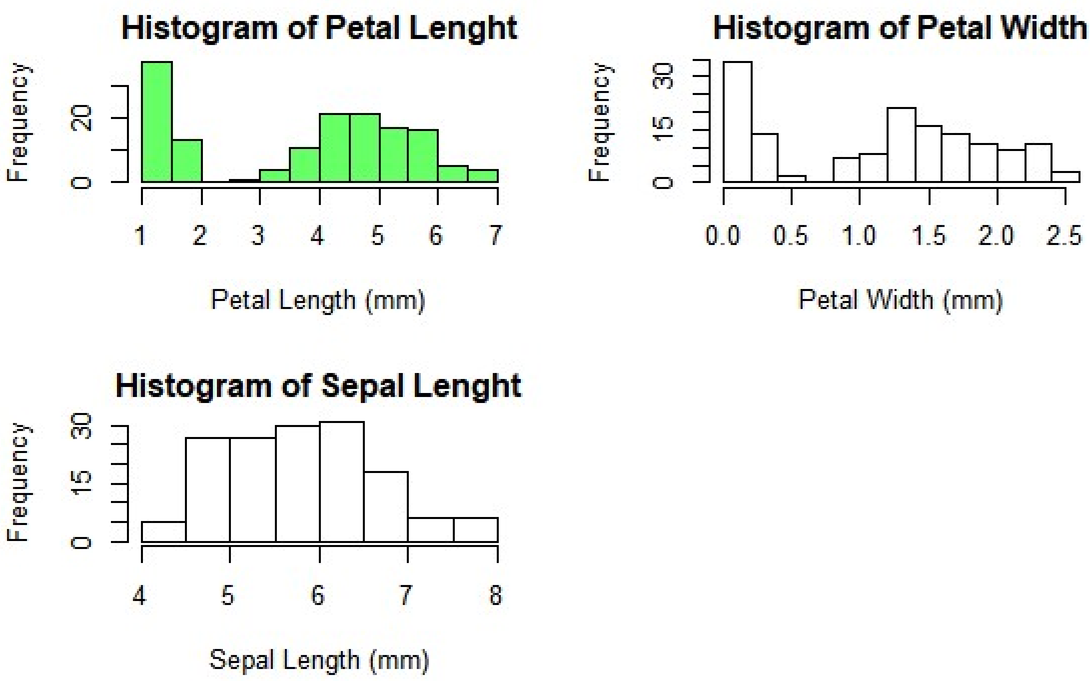
Make histogram for the numerical features, to see how they distribute

Hide

```
par(mfrow=c(2,2))
hist(iris$Petal.Length,main='Histogram of Petal Lenght',xlab='Petal Length (mm)',col = '#66FF66')
hist(iris$Petal.Width,main='Histogram of Petal Width',xlab='Petal Width (mm)')
```

Hide

```
hist(iris$Sepal.Length,main='Histogram of Sepal Lenght',xlab='Sepal Length (mm)')
hist(iris$Petal.Length,main='Histogram of Sepal Width',xlab='Sepal Width (mm)')
```



(3) Table

Make table for the categorical features, to see how they distribute

Hide

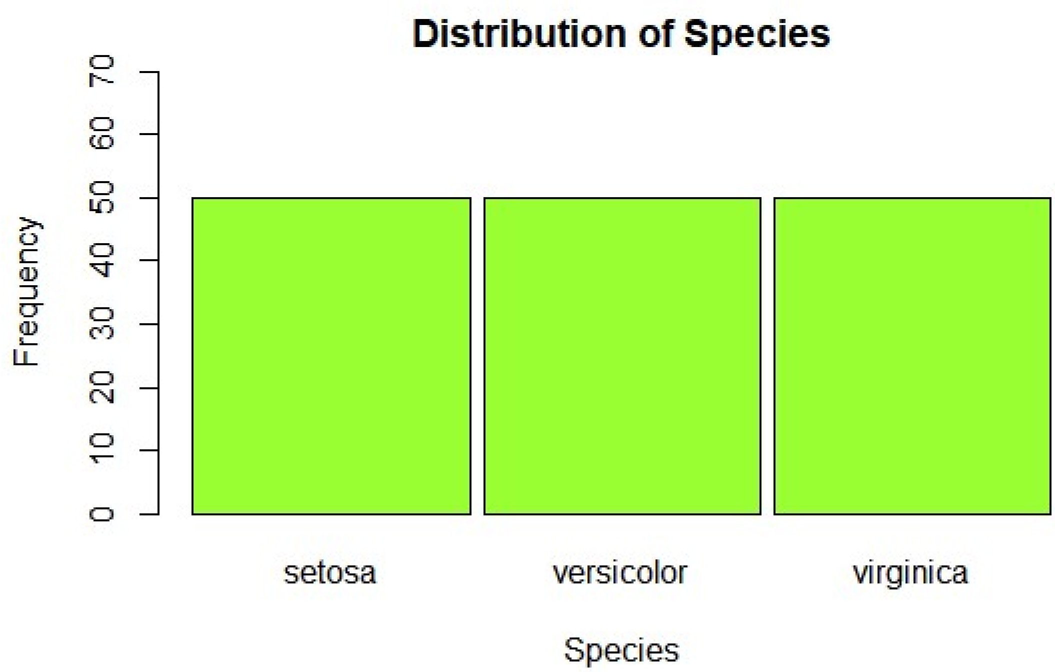
```
table(iris$Species)
```

```

    setosa versicolor  virginica 
        50         50         50 
```

Hide

```
barplot(table(iris$Species), main = 'Distribution of Species', xlab = 'Species', ylab
= 'Frequency',space = .05, col = '#99ff33', ylim = c(0,70) )
```



Hide

```
NA
NA
```

2. **Rivers**

The data set rivers contains the lengths (in miles) of 141 major rivers in North America.

(1) What proportion are less than 500 miles long?

Hide

```
t = length(rivers[rivers < 500])
p = t/length(rivers)
p
```

```
[1] 0.5815603
```

(2) What proportion are less than the mean length?

Hide

```
t = length(rivers[rivers < mean(rivers)])
p = t/length(rivers)
p
```

```
[1] 0.6666667
```

(3) What is the 0.75 quantile?

Hide

```
third = quantile(rivers)
third
```

```
  0%  25%  50%  75% 100% 
 135  310  425  680 3710 
```

The 0.75 quantile is 680miles.

3. **pi2000**

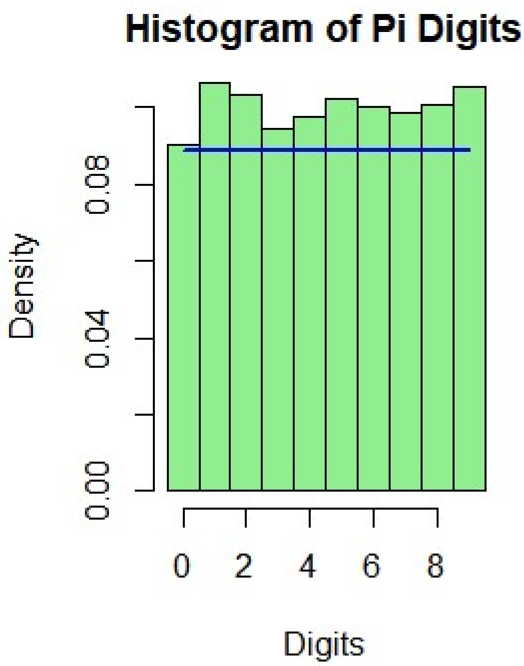
Fit a density estimate to the data set pi2000 (UsingR). Compare with the appropriate histogram. Why might you want to add an argument like breaks =0:10-0.5 to hist()?

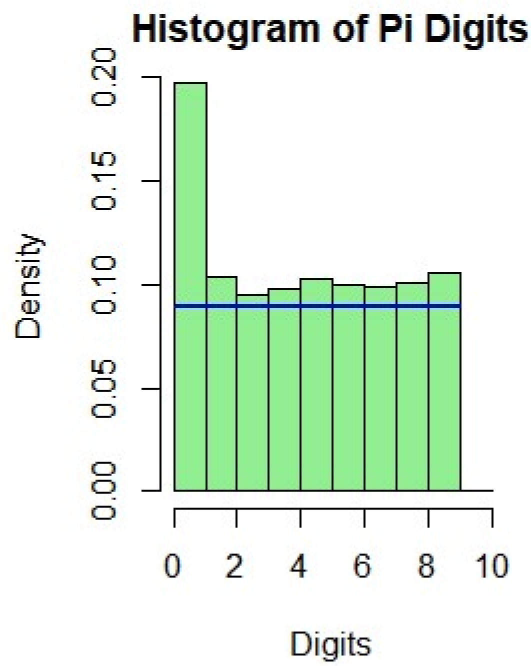
Hide

```
data(pi2000)
par(mfrow=c(1,2))
x <- pi2000
h <- hist(x, breaks = 0:10-0.5, col="light green", xlab='Digits', main='Histogram of P
i Digits',prob=T)
xfit <- seq(min(x),max(x),length=9)
yfit <- dunif(xfit,0,9)
yfit <- yfit*.8
lines(xfit,yfit,col='blue',lwd=2)
```

Hide

```
h <- hist(x, breaks = 0:10, col="light green", xlab='Digits', main='Histogram of Pi Di
gits',prob=T)
xfit <- seq(min(x),max(x),length=9)
yfit <- dunif(xfit,0,9)
yfit <- yfit*.8
lines(xfit,yfit,col='blue',lwd=2)
```





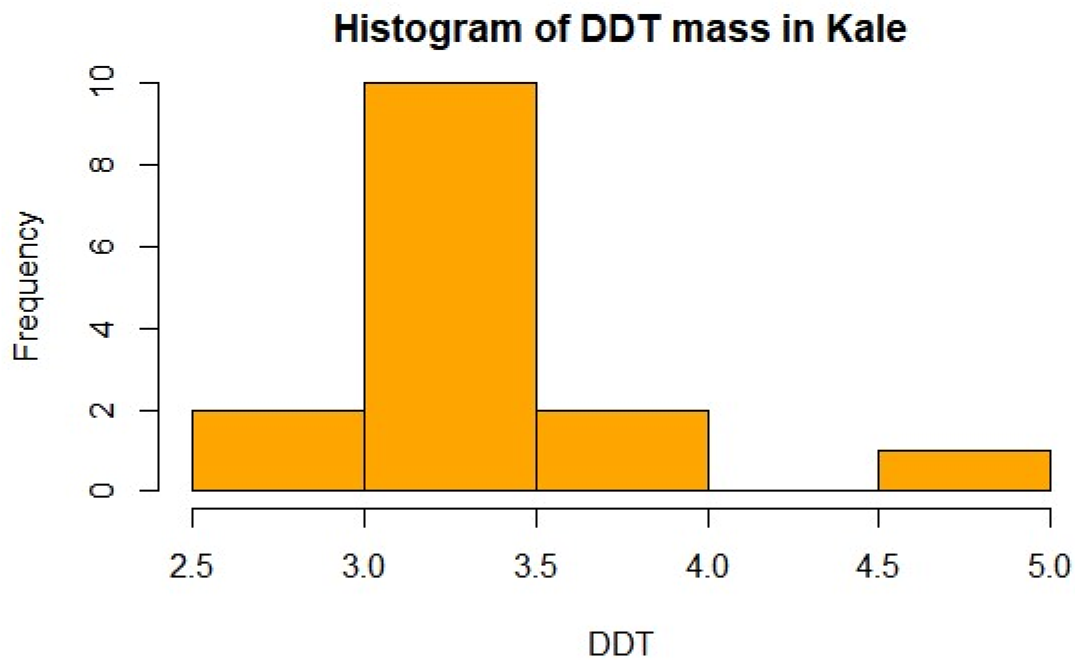
Using breaks = 0:10-0.5 helps display the histogram more accurately since using breaks = 10 for example, causes the histogram to be plotted out of proportion.

4. **MASS**

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.

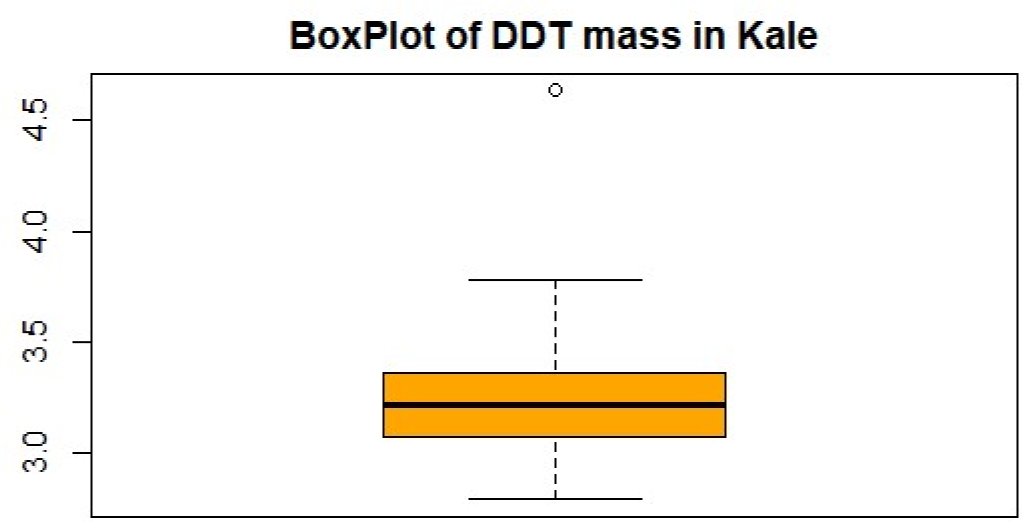
Hide

```
hist(DDT,breaks=5,col="orange", main="Histogram of DDT mass in Kale")
```



Hide

```
boxplot(DDT,col="orange",main="BoxPlot of DDT mass in Kale")
```



MASS Estimates

$mean \approx 3.6$

$sd \approx 0.4$

MASS Actuals

Hide

```
str(DDT)
```

```
 num [1:15] 2.79 2.93 3.22 3.78 3.22 3.38 3.18 3.33 3.34 3.06 ...
```

Hide

```
sprintf('Mean mass is %s ', mean(DDT))
```

```
[1] "Mean mass is 3.328 "
```

Hide

```
sprintf('Standard deviation is %.2f ', sd(DDT))
```

```
[1] "Standard deviation is 0.44 "
```

5. Two Graphics

It can be illuminating to view two different graphics of the same data set at once. A simple way to stack graphics is to specify that a figure will contain two graphics by using the command

> par(mfrow=c(2,1)

> hist(x)

> boxplot(x, horizontal=TRUE)

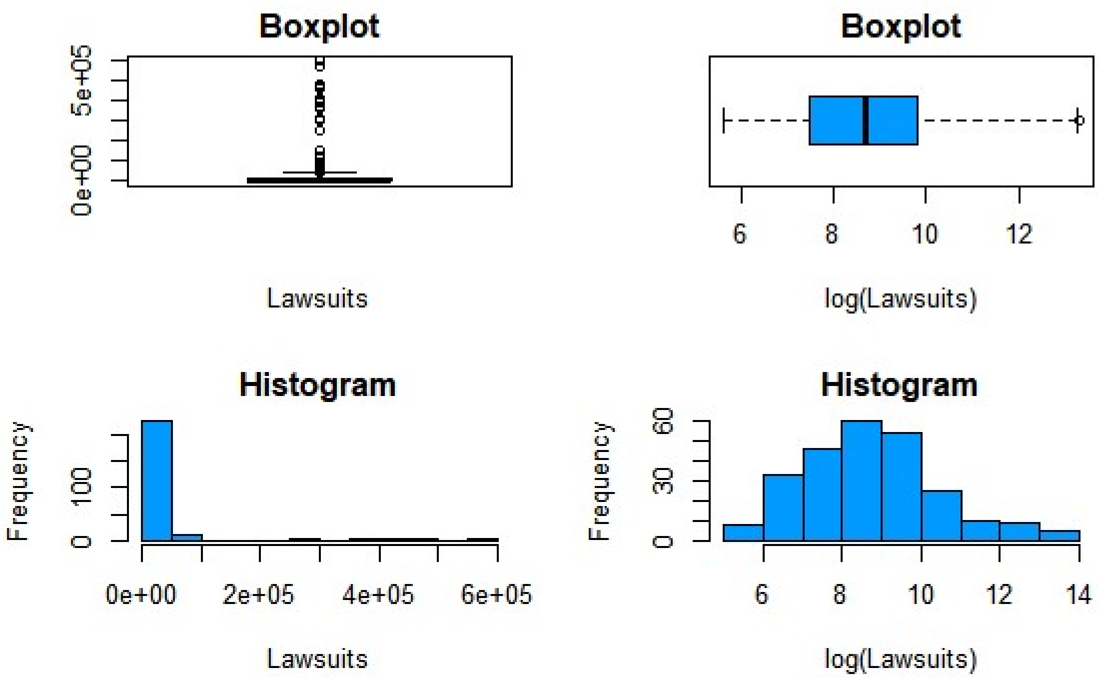
will produce stacked graphics. (The graphics device will remain divided until you change it back with a command such as par (mfrow=c(1, 1)) or close the device.) For the data set lawsuits (UsingR), make stacked graphics of lawsuits and log (lawsuits). Could you have guessed where the middle 50% of the data would have been without the help of the boxplot?

Hide

```
par(mfrow=c(2,2))
x <- (lawsuits)
boxplot(x, col="#0099ff", main='Boxplot', xlab='Lawsuits')
boxplot(log(x), col='#0099ff', horizontal=TRUE, main='Boxplot',
xlab='log(Lawsuits)')
```

Hide

```
hist(x, col="#0099ff", main='Histogram', xlab='Lawsuits')
hist(log(x), col='#0099ff', main='Histogram',
xlab='log(Lawsuits)')
```



Determining the median of the data would have been impossible without the aid of box plot of log(lawsuits).

6. **Sex, Age, and Smoking**

Let sex = c(1,1,1,1,2,2,2,2,2,2); smoking = c(1,0,1,0,1,0,0,0,1,1); age=c(31:40) in R. A data frame is constructed as zz = data.frame(sex, smoking, age). Give the results of following R commands: 1) apply(zz[-1,], 2, min)

Hide

```
sex <- c(1,1,1,1,2,2,2,2,2,2)
smoking <- c(1,0,1,0,1,0,0,0,1,1)
age <- c(31:40)
zz <- data.frame(sex, smoking, age)
zz
```

sex <dbl>	smoking <dbl>	age <int>
1	1	31
1	0	32
1	1	33
1	0	34
2	1	35
2	0	36
2	0	37
2	0	38
2	1	39
2	1	40

1-10 of 10 rows

Hide

```
apply(zz[-1,], 2, min)
```

```
    sex smoking     age
      1       0      32
```

Hide

```
zz[zz[,3]>35,]
```

	sex <dbl>	smoking <dbl>	age <int>
6	2	0	36
7	2	0	37
8	2	0	38

	sex	smoking	age
	<dbl>	<dbl>	<int>
9	2	1	39
10	2	1	40

5 rows

Hide

```
zz[order(zz["smoking"], zz["age"]),  ]
```

	sex	smoking	age
	<dbl>	<dbl>	<int>
2	1	0	32
4	1	0	34
6	2	0	36
7	2	0	37
8	2	0	38
1	1	1	31
3	1	1	33
5	2	1	35
9	2	1	39
10	2	1	40

1-10 of 10 rows

Hide

```
subset(zz, zz["sex"]==1)
```

	sex	smoking	age
	<dbl>	<dbl>	<int>
1	1	1	31
2	1	0	32
3	1	1	33
4	1	0	34

4 rows

Hide

```
tapply(zz$age, zz$smoking, max)
```

```
 0  1 
38 40 
```

Hide

```
apply(zz[,-3], 1, function(x){ sum(x) })
```

```
 [1] 2 1 2 1 3 2 2 2 3 3
```