Homework 01 Solutions

Alan Mishler
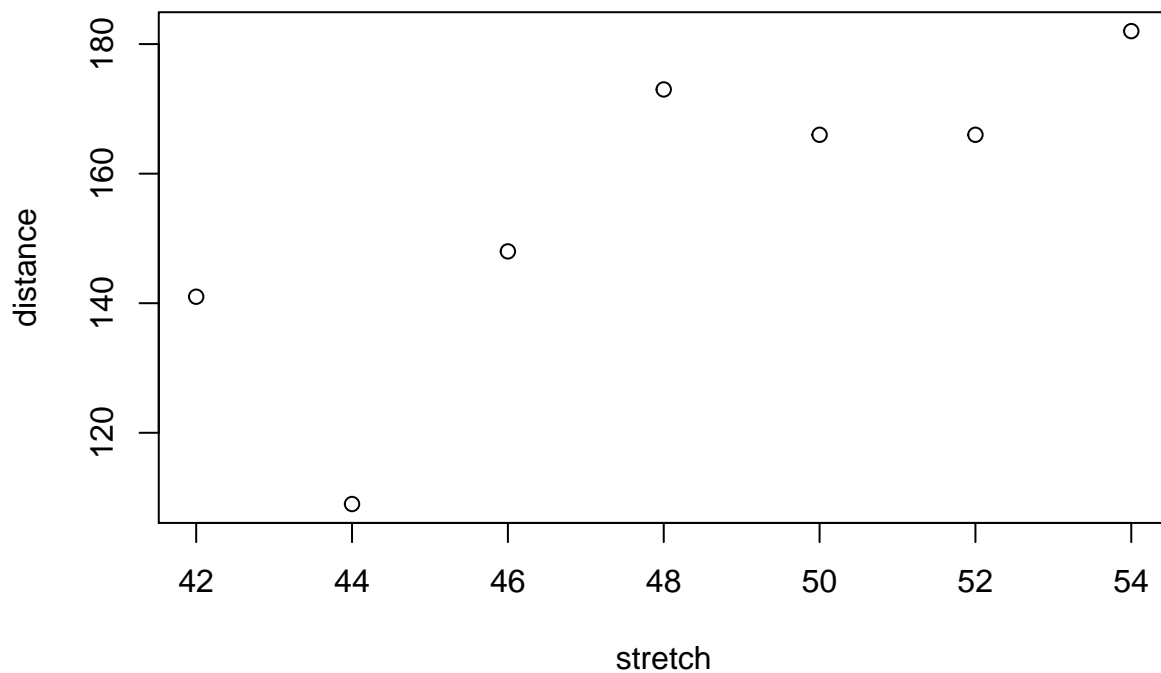
8/30/2018

Chapter 1 Exercises

```r
load("../usingR.RData")
```
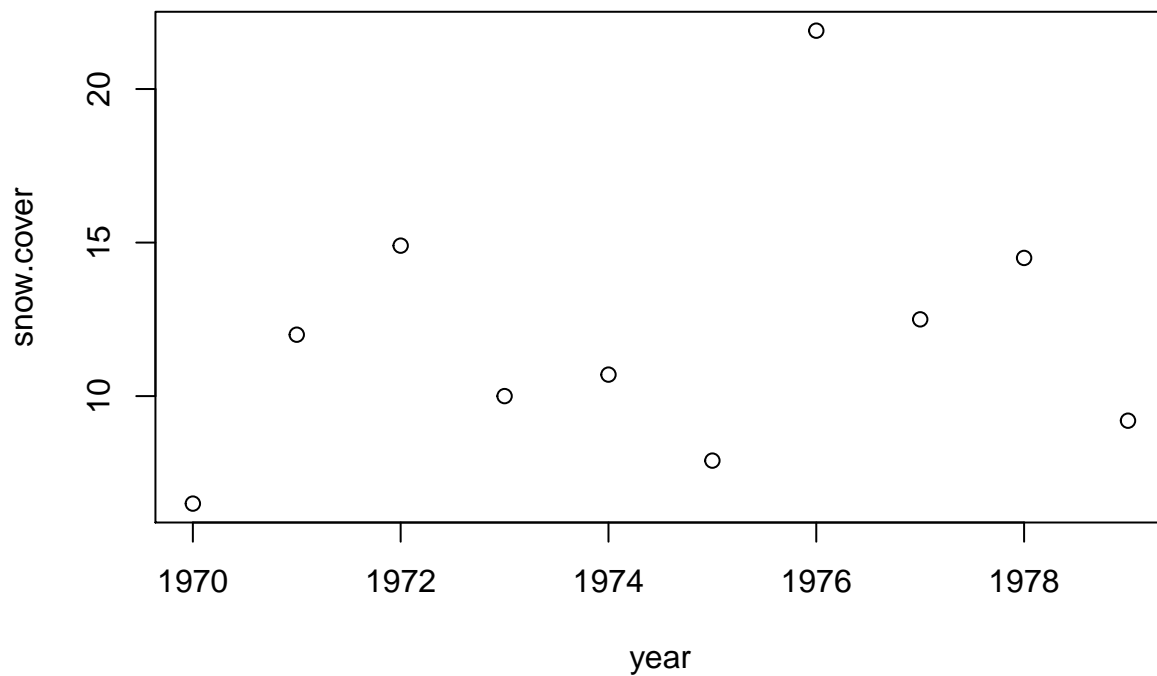
Exercise 1

```r
plot(distance ~ stretch, data = elasticband)
```
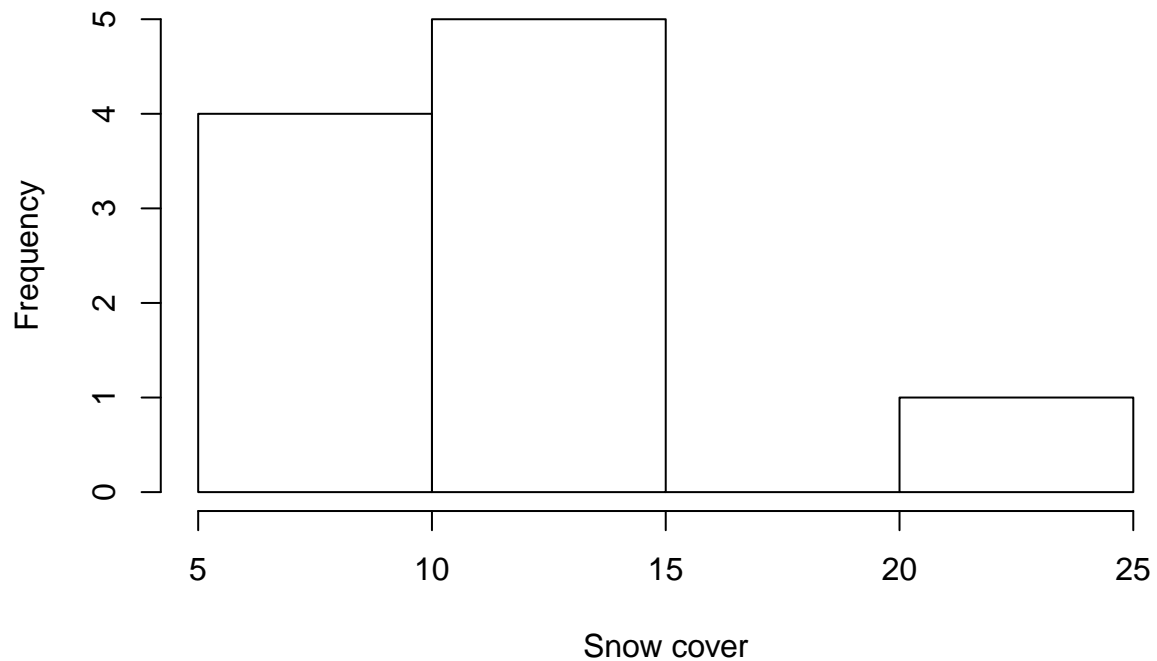

Exercise 2

```r
# i.
snow <- data.frame(year = 1970:1979,
                   snow.cover = c(6.5, 12, 14.9, 10, 10.7, 7.9, 21.9, 12.5, 14.5, 9.2))

# ii.
plot(snow.cover ~ year, data = snow)
```
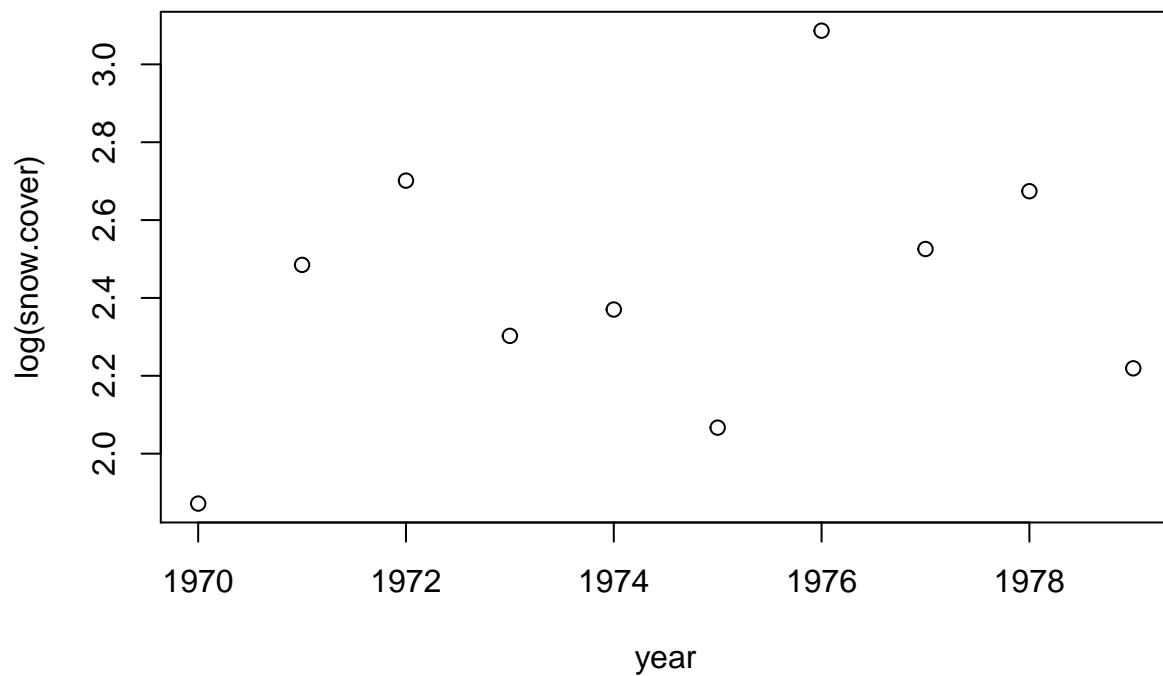
Histogram of snow cover

```r
hist(log(snow$snow.cover), xlab = "Snow cover", main = "Histogram of snow cover")
```

Histogram of snow cover

Exercise 3
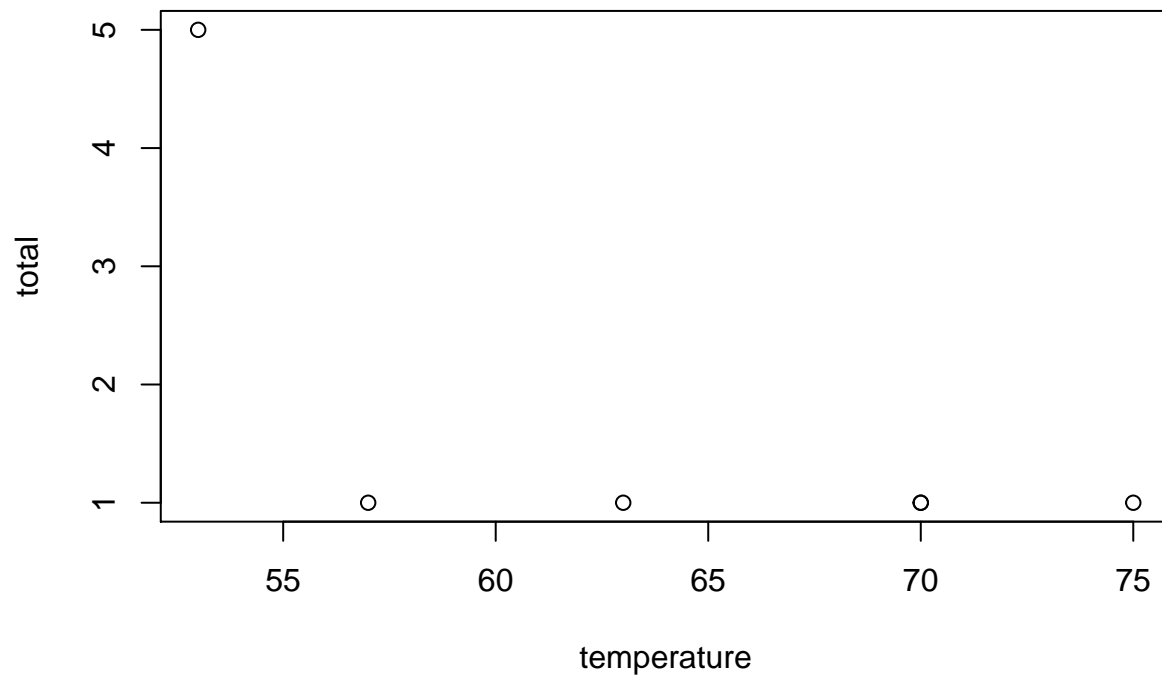
```r
orings <- data.frame(temperature = c(53, 57, 63, 70, 70, 75),
                     erosion = c(3, 1, 1, 1, 1, 0),
```

```
              blowby = c(2, 0, 0, 0, 0, 2),
              total = c(5, 1, 1, 1, 1, 1))

plot(total ~ temperature, data = orings)
```


Chapter 2 Exercises

Exercise 1

```
# 2.1(a)
answer <- 0
for (j in 3:5) {answer <- j + answer}
print(answer)
```

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

```
# 2.1(b)
answer <- 10
for (j in 3:5) {answer <- j + answer}
print(answer)
```

```
## [1] 22
```

```
# 2.1(c)
answer <- 10
for (j in 3:5) {answer <- j*answer}
print(answer)
```

```
## [1] 600
```

Exercise 2

```
?prod
prod(c(10, 3, 4, 5))
```

```
## [1] 600
```

Exercise 5

```
radii <- 3:20
spheres <- data.frame(radius = radii, volume = 4*pi*radii^3)
spheres
```

```
##    radius      volume
## 1       3    339.2920
## 2       4    804.2477
## 3       5   1570.7963
## 4       6   2714.3361
## 5       7   4310.2651
## 6       8   6433.9818
## 7       9   9160.8842
## 8      10  12566.3706
## 9      11  16725.8393
## 10     12  21714.6884
## 11     13  27608.3162
## 12     14  34482.1210
## 13     15  42411.5008
## 14     16  51471.8540
## 15     17  61738.5788
## 16     18  73287.0734
## 17     19  86192.7360
## 18     20 100530.9649
```

Exercise 6

```
# Identify which columns are factors
factor_columns <- sapply(tinting, is.factor)

# Get levels of factor columns
sapply(tinting[factor_columns], levels)
```

```
## $sex
## [1] "f" "m"
##
## $tint
## [1] "no" "lo" "hi"
##
## $target
## [1] "locon" "hicon"
##
## $agegp
## [1] "Younger" "Older"
```

```r
# Check which columns are ordered factors
sapply(tinting, is.ordered)
```
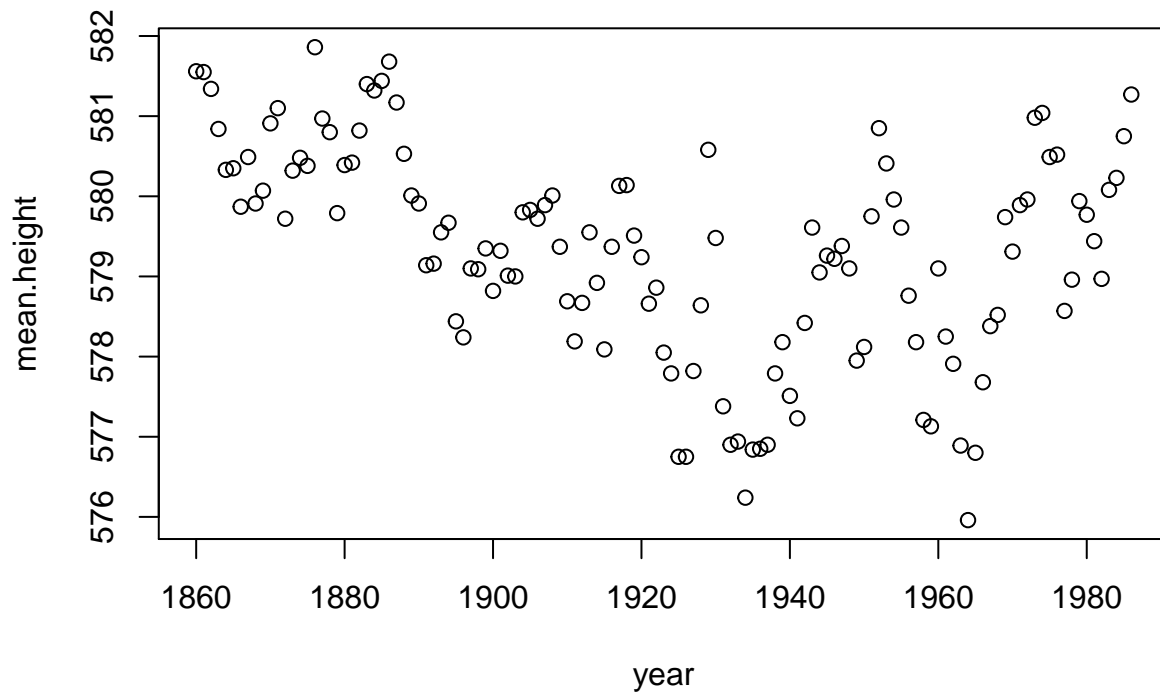
```
##   case     id    age    sex   tint target     it   csoa  agegp
## FALSE  FALSE  FALSE  FALSE   TRUE  FALSE  FALSE  FALSE  FALSE
```

Chapter 3 Exercises

Exercise 1
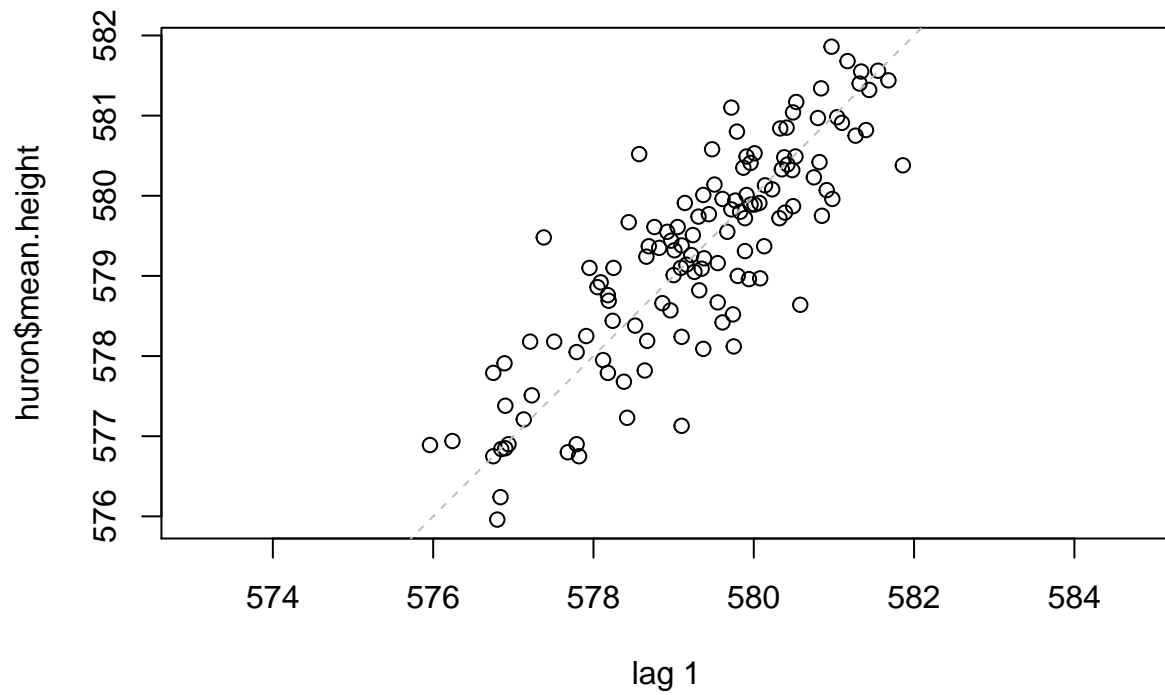
```r
# 3.1(a)
plot(mean.height ~ year, data = huron)

# 3.1(b)
# Note: These labels won't necessarily show up in your saved output, which is fine.
identify(huron$year, huron$mean.height, labels = huron$year)
```



```
## integer(0)
```
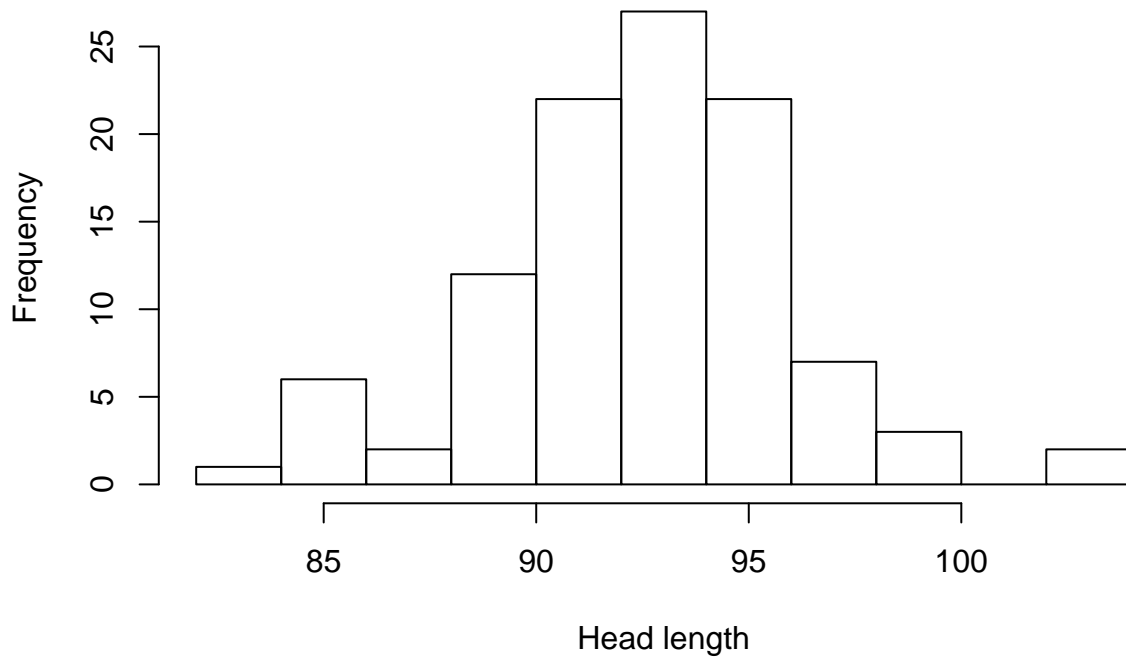
```r
# 3.1(c)
# Note: If you read the docs for lag.plot(), with ?lag.plot or help(lag.plot),
# you'll see that you can't change certain graphical parameters, including xlab and ylab.
lag.plot(huron$mean.height, do.lines = FALSE)
```

Exercise 3

```
# 3.3(a)
hist(possum$hdlngth, xlab = "Head length")
```
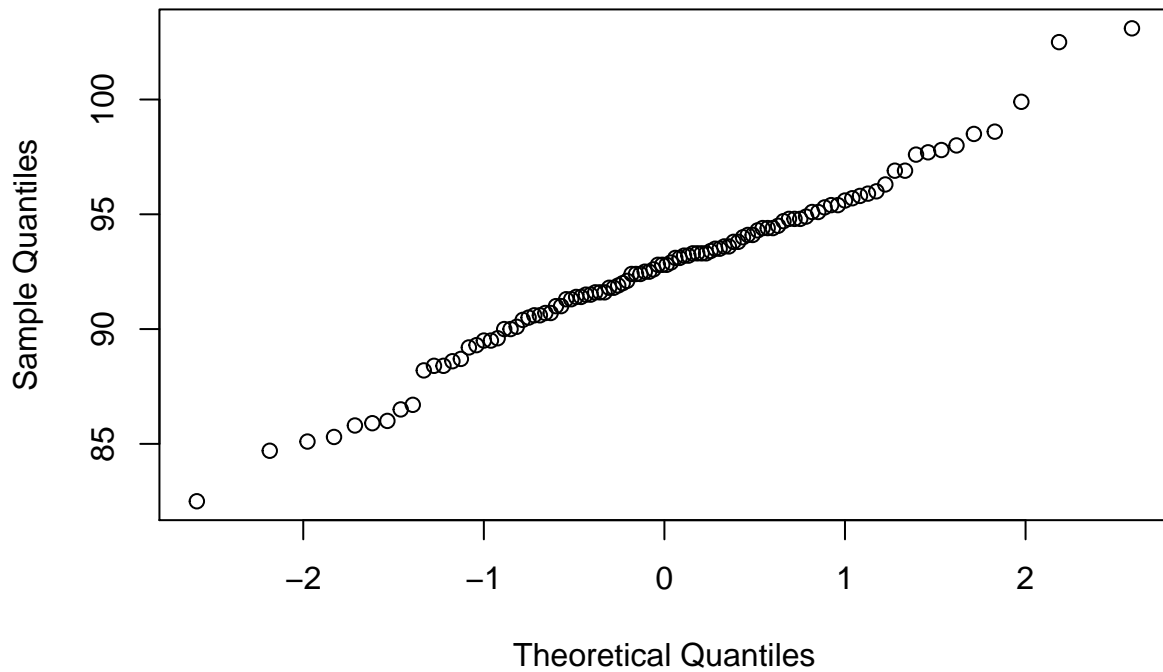
Histogram of possum$hdlngth

```r
# 3.3(b)
# Note: the code in the problem is incorrectly given as stem(qqnorm(possum$hdlength))
stem(possum$hdlngth)
```

```
## 
##   The decimal point is at the |
## 
##     82 | 5
##     83 |
##     84 | 7
##     85 | 1389
##     86 | 057
##     87 |
##     88 | 24467
##     89 | 23556
##     90 | 001456677
##     91 | 00334455666889
##     92 | 014445568889
##     93 | 112233334556688
##     94 | 0113444578889
##     95 | 113446789
##     96 | 0399
##     97 | 678
##     98 | 056
##     99 | 9
##    100 |
##    101 |
##    102 | 5
##    103 | 1
```
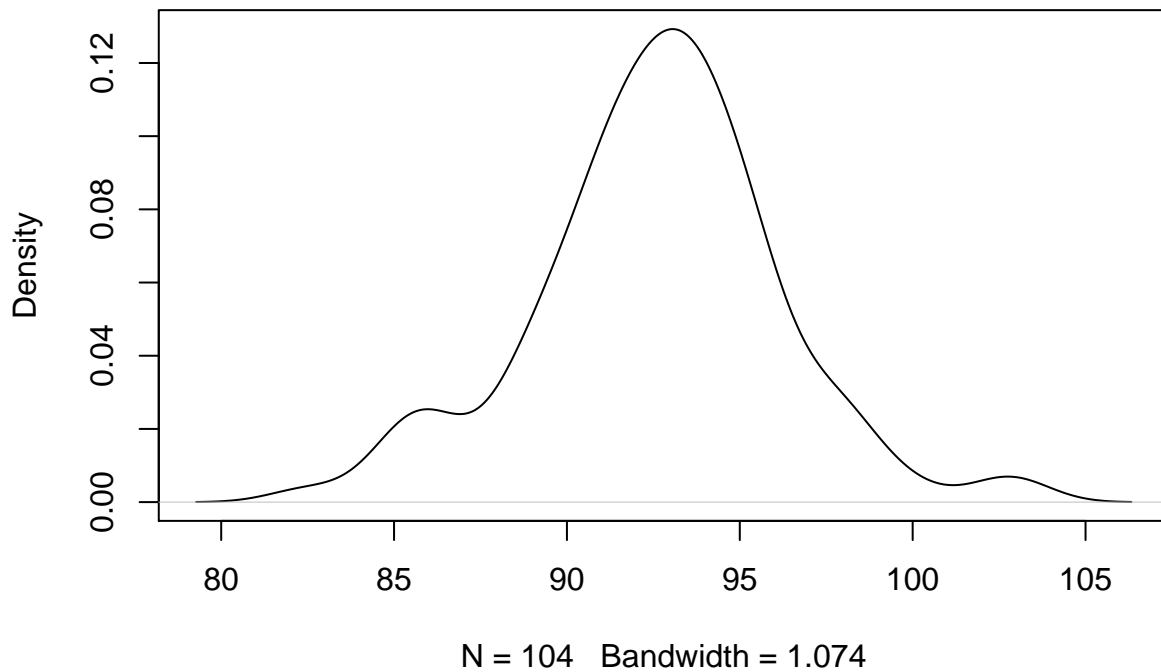
```r
# 3.3(c)
qqnorm(possum$hdlngth)
```

Normal Q-Q Plot


```
# 3.3(d)
plot(density(possum$hdlngth), main = "Estimated density of possum head length")
```

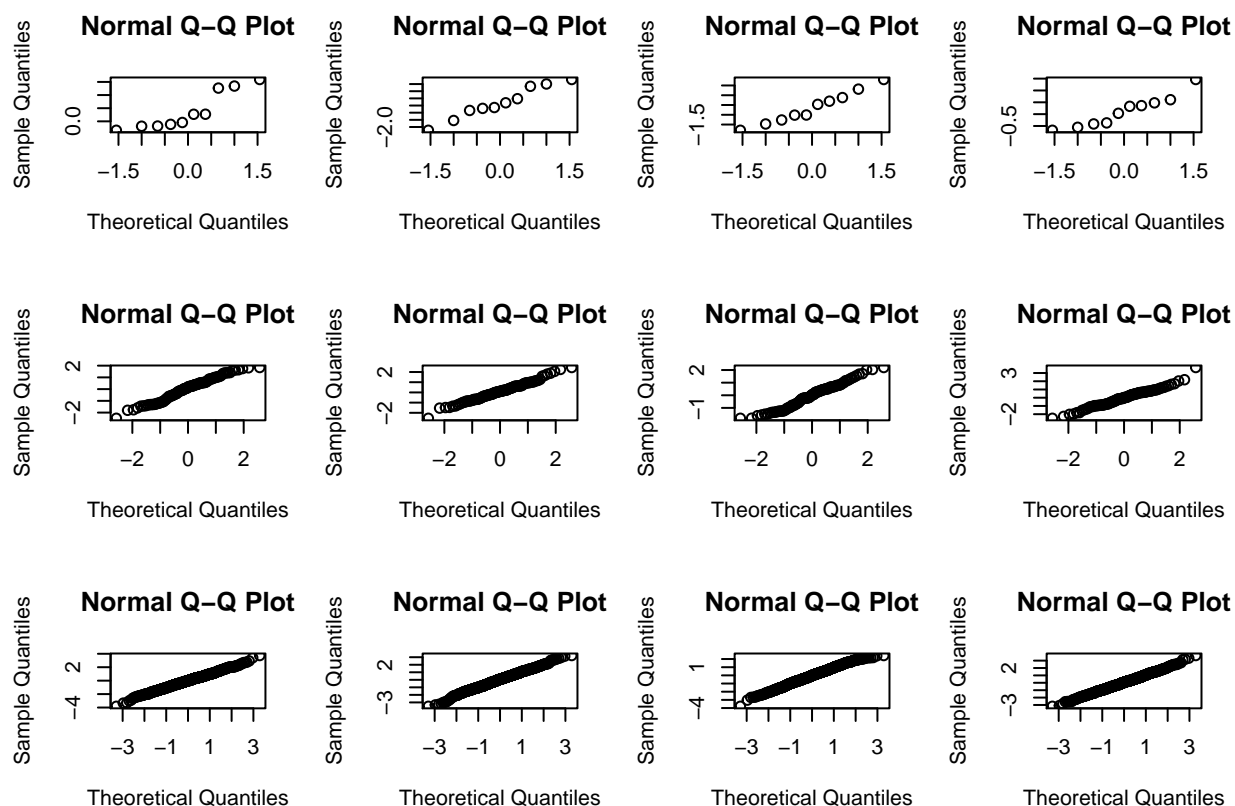## Estimated density of possum head length

N = 104   Bandwidth = 1.074

These displays provide slightly different types of information. The stem and leaf plot is essentially a sideways histogram, with the bin widths fixed at 1. It allows you to see all the individual data points. The histogram

provides more flexibility in that bin widths are adjustable, but it is sensible to the bin width chosen. The density plot is a smoothed version of the histogram, which may give a better sense of the shape of the distribution, but it is sensitive to the parameters of kernel used to estimate the curve. The normal QQ plot makes it easy to see deviations from normality, but it is hard to see what the distribution is if it is not normal. The other displays make it much easier to see the shape of the distribution.

Exercise 5

```r
par(mfrow = c(3, 4))
for (i in 1:4) {
  qqnorm(rnorm(10))
}
for (i in 1:4) {
  qqnorm(rnorm(100))
}
for (i in 1:4) {
  qqnorm(rnorm(1000))
}
```



```r
# Typically you'll want to reset the grid after changing it.
par(mfrow = c(1, 1))
```

The larger the sample size, the more representative the sample (i.e. the closer to normal), and hence the straighter the line in the QQ plot.