Homework 01 Solutions

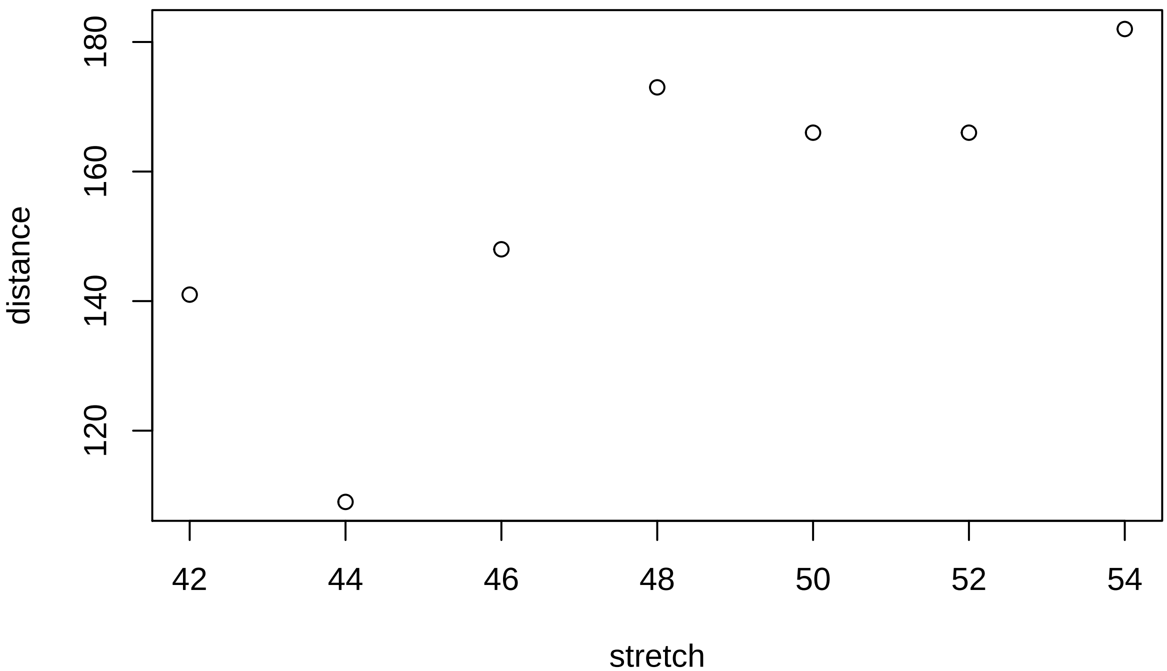
8/30/2020

Chapter 1 Exercises

```
load("../usingR.RData")
```

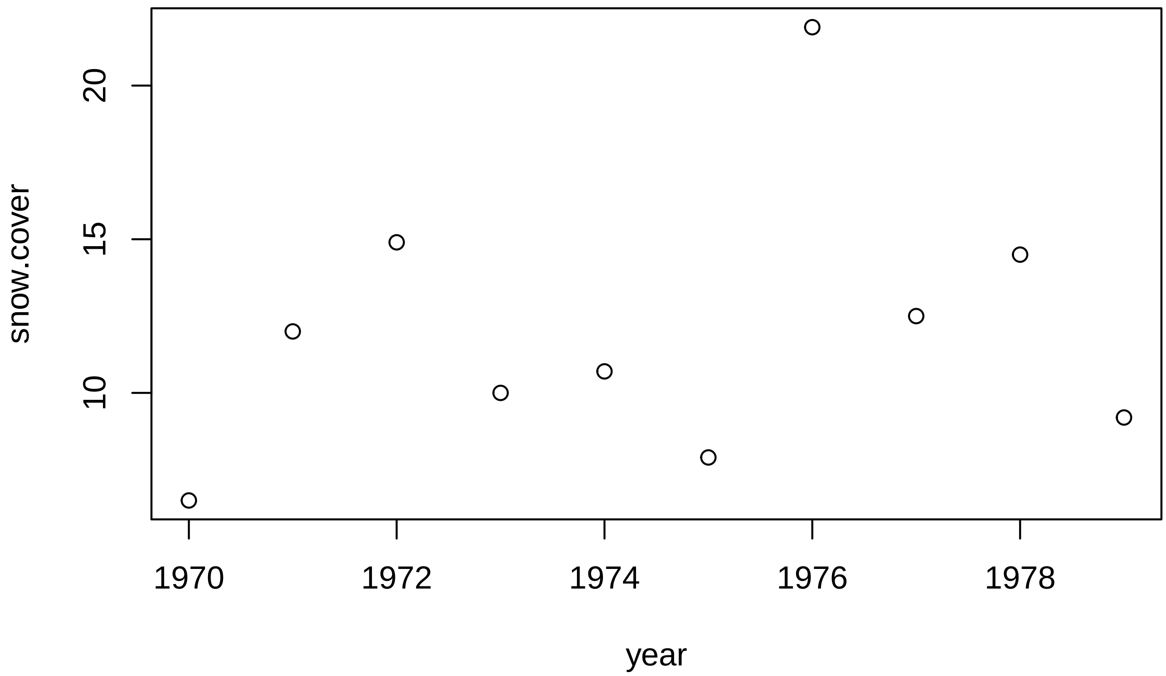
Exercise 1

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

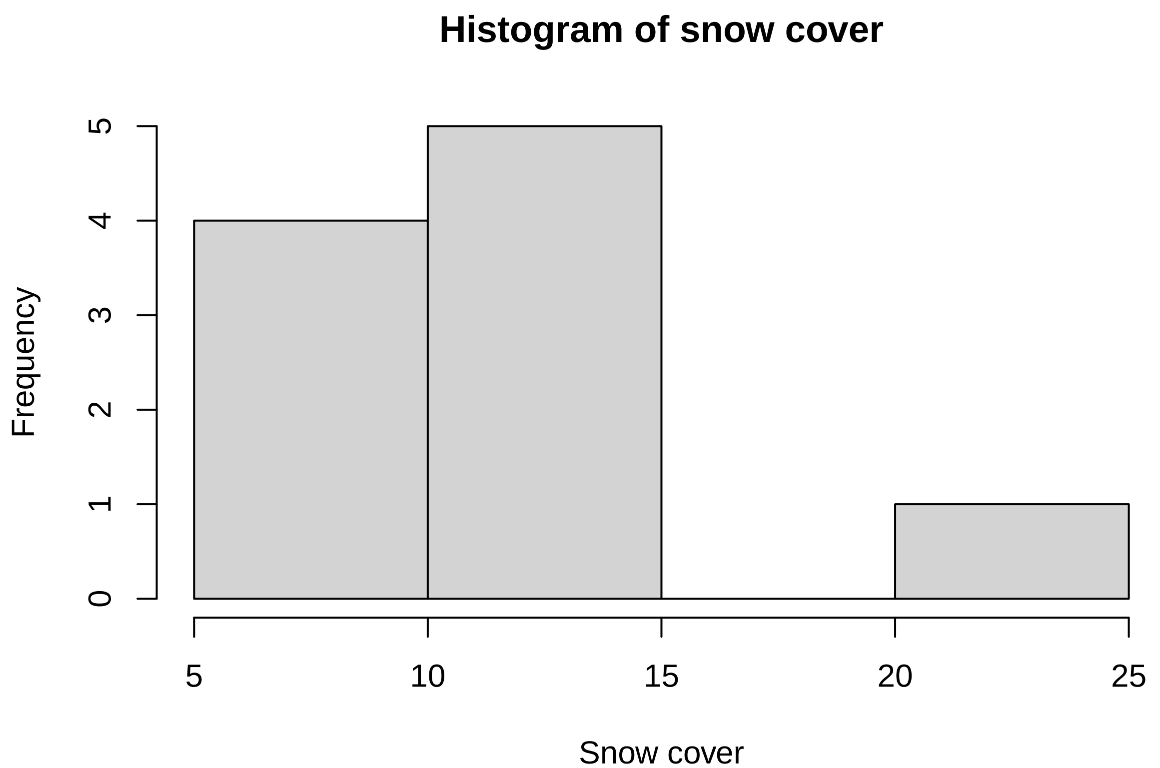


Exercise 2

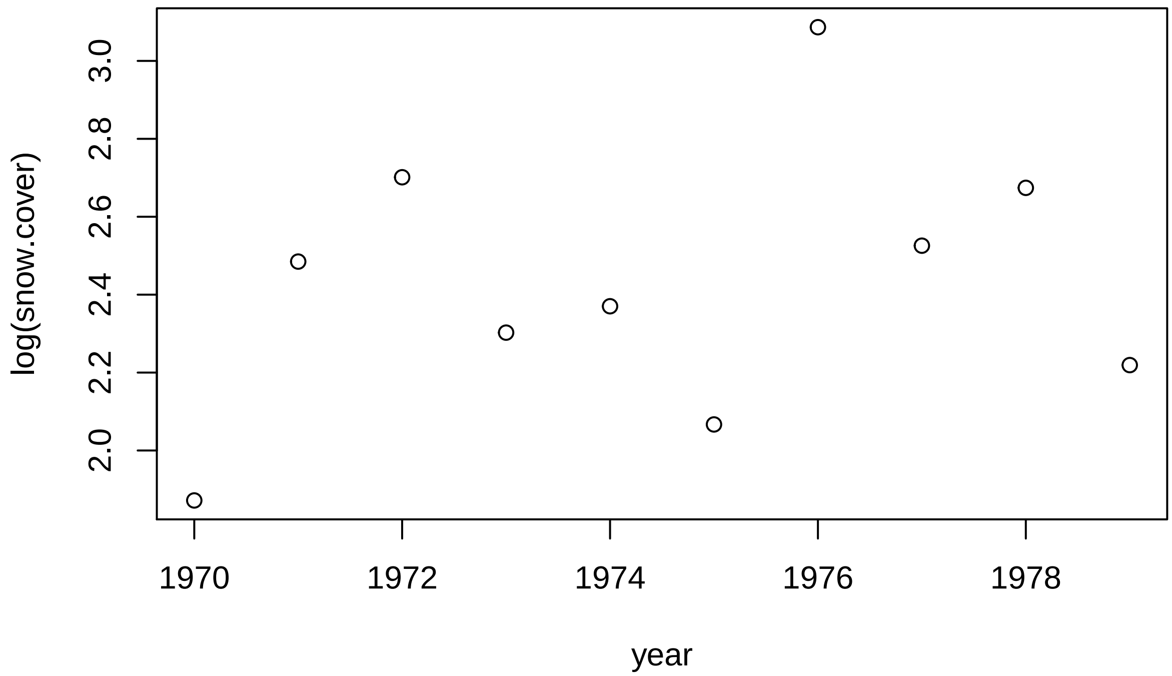
```
# i.
snow <- data.frame(year = 1970:1979,
                   snow.cover = c(6.5, 12, 14.9, 10, 10.7, 7.9, 21.9, 12.5, 14.5, 9.2))

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



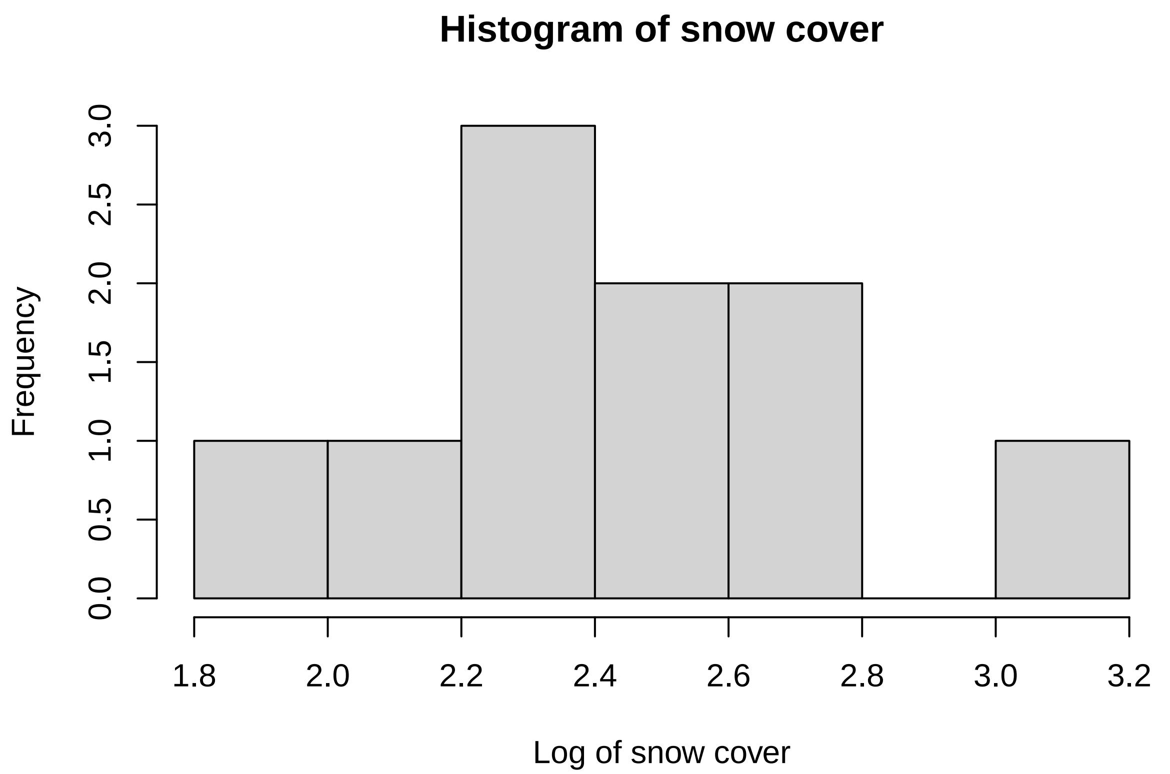

```
# iii.
hist(snow$snow.cover, xlab = "Snow cover", main = "Histogram of snow cover")
```




```
# iv.
plot(log(snow.cover) ~ year, data = snow)
```



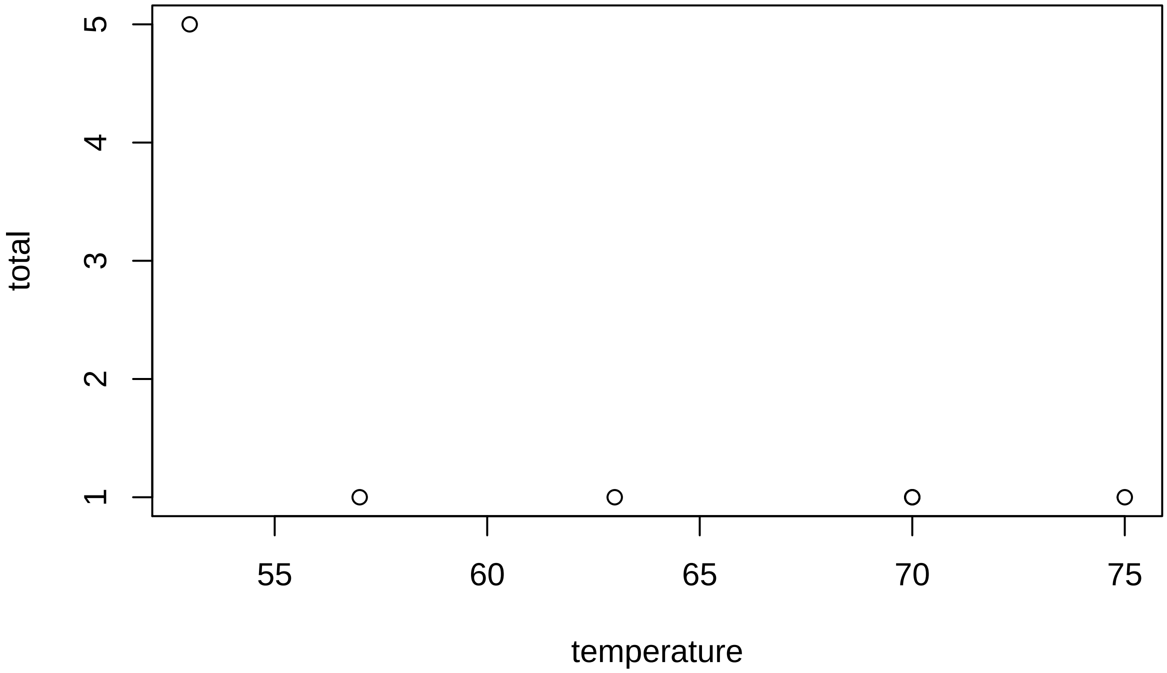

```
hist(log(snow$snow.cover), xlab = "Log of snow cover", main = "Histogram of snow cover")
```



Exercise 3

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

```r
                    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

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

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

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

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

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

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

Exercise 2

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

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

Exercise 5

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

```
##    radius     volume
## 1       3   113.0973
## 2       4   268.0826
## 3       5   523.5988
## 4       6   904.7787
## 5       7  1436.7550
## 6       8  2144.6606
## 7       9  3053.6281
## 8      10  4188.7902
## 9      11  5575.2798
## 10     12  7238.2295
## 11     13  9202.7721
## 12     14 11494.0403
## 13     15 14137.1669
## 14     16 17157.2847
## 15     17 20579.5263
## 16     18 24429.0245
## 17     19 28730.9120
## 18     20 33510.3216
```

When you type `pi` in R and hit `Enter`, R prints the value of π to some number of digits.

Exercise 6

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

```
##   case     id    age    sex   tint target     it   csoa  agegp
##  FALSE  FALSE  FALSE   TRUE   TRUE   TRUE  FALSE  FALSE   TRUE
```

```r
# 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"
```

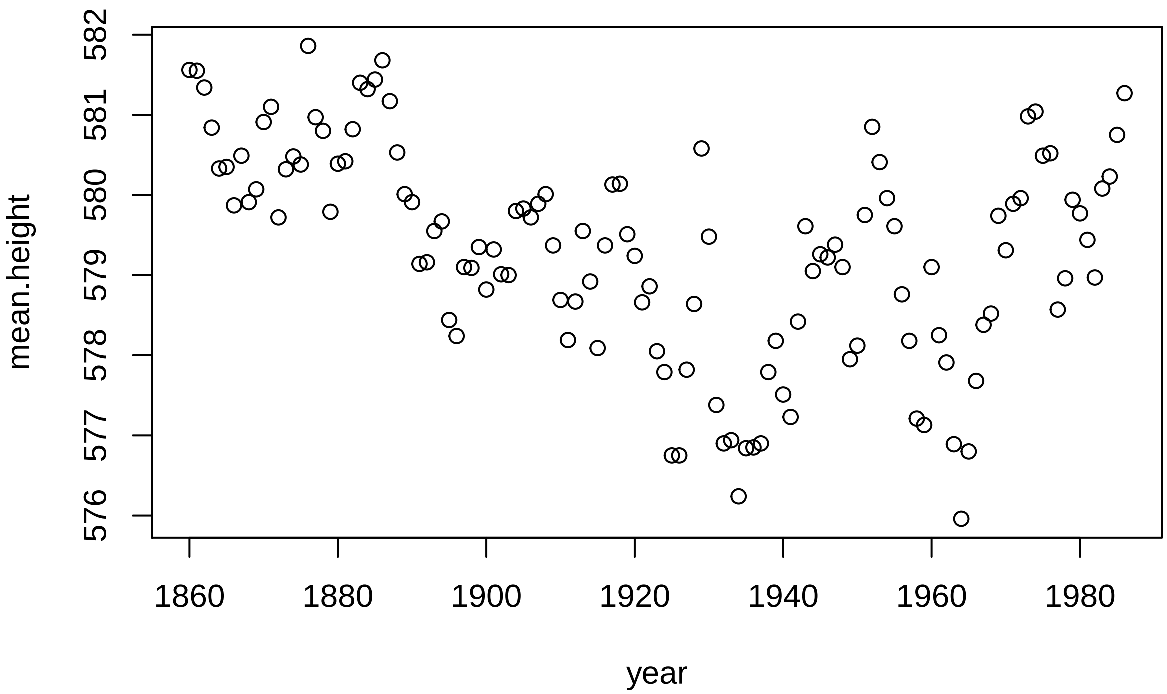
```r
# Check which columns are ordered factors
sapply(tinting, is.ordered)
```

```
##   case     id    age    sex   tint target     it   csoa  agegp
##  FALSE  FALSE  FALSE  FALSE   TRUE  FALSE  FALSE  FALSE  FALSE
```

Chapter 3 Exercises

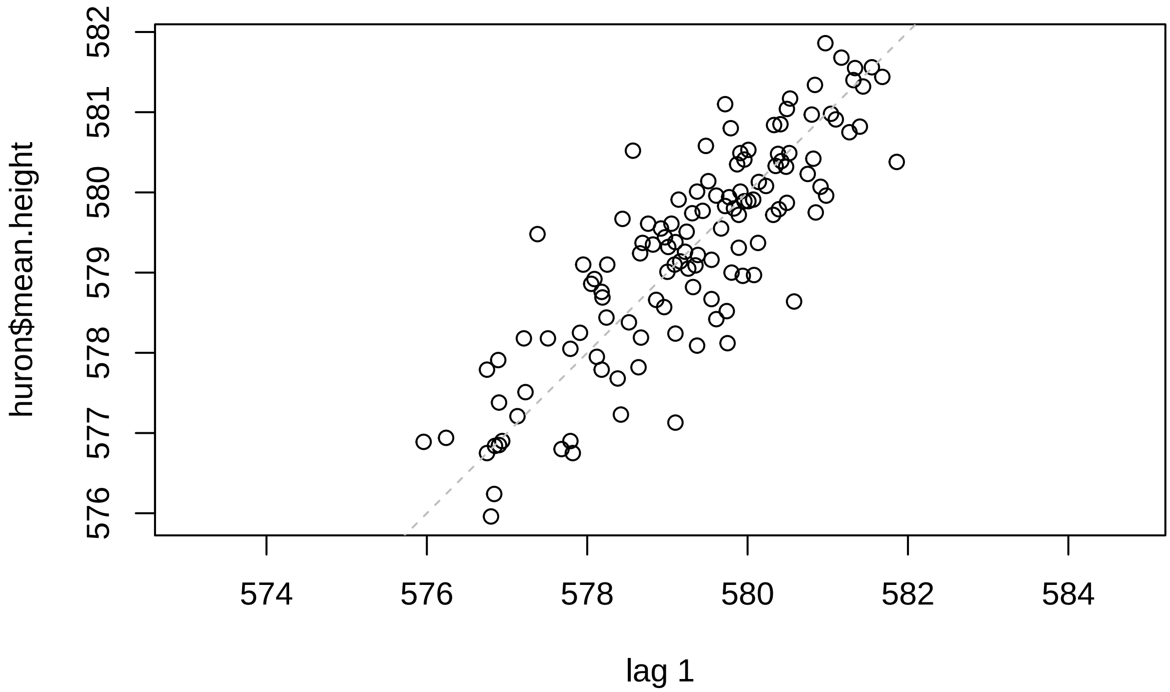
Exercise 1

```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)
```



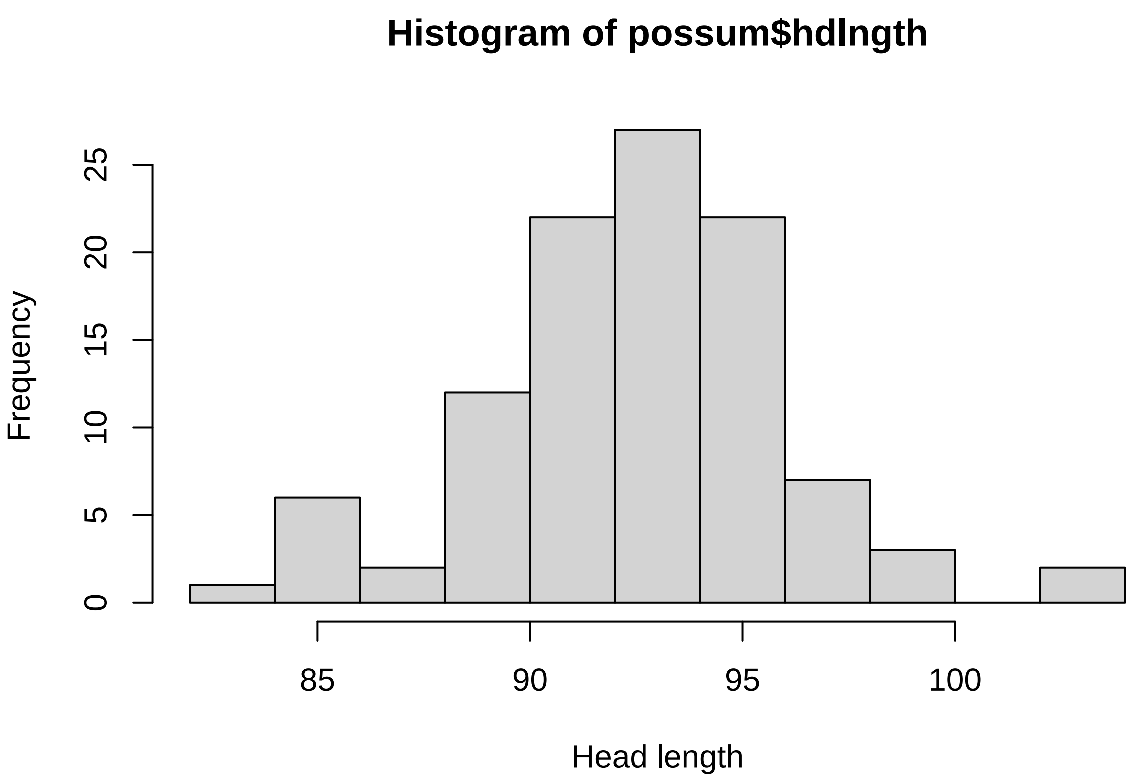

```
## integer(0)
```

```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

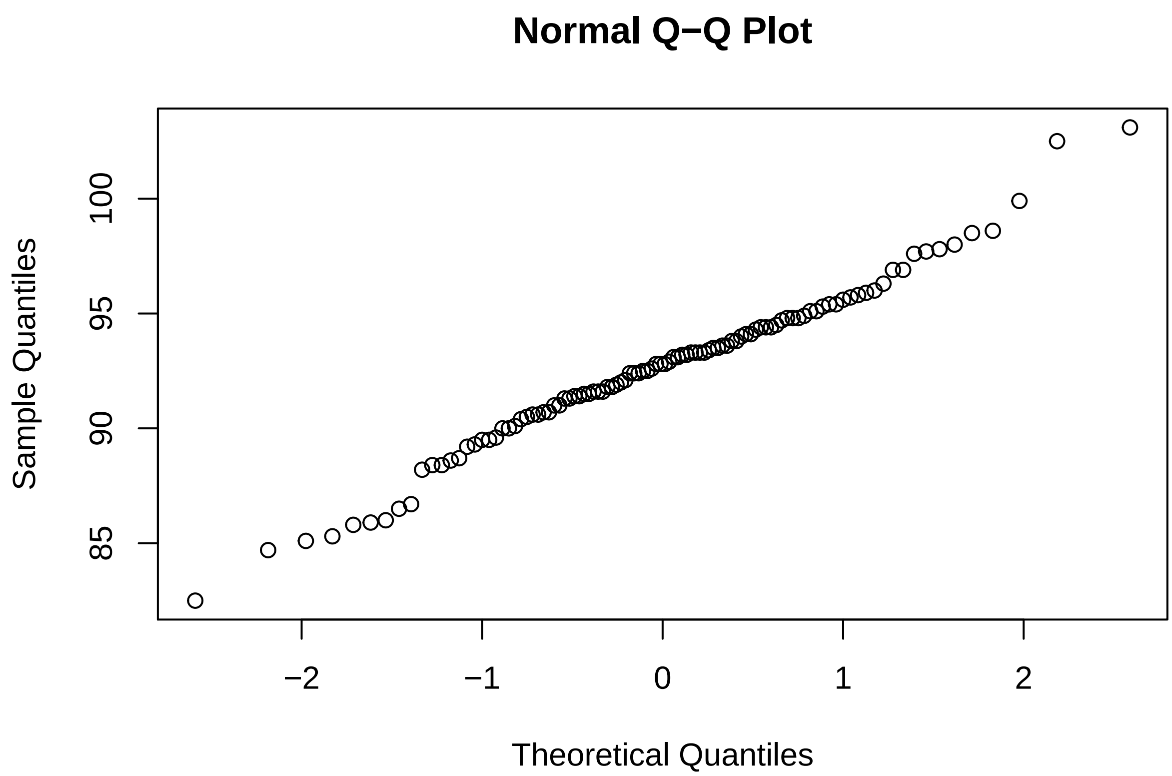
```
# 3.3(a)
hist(possum$hdlngth, xlab = "Head length")
```



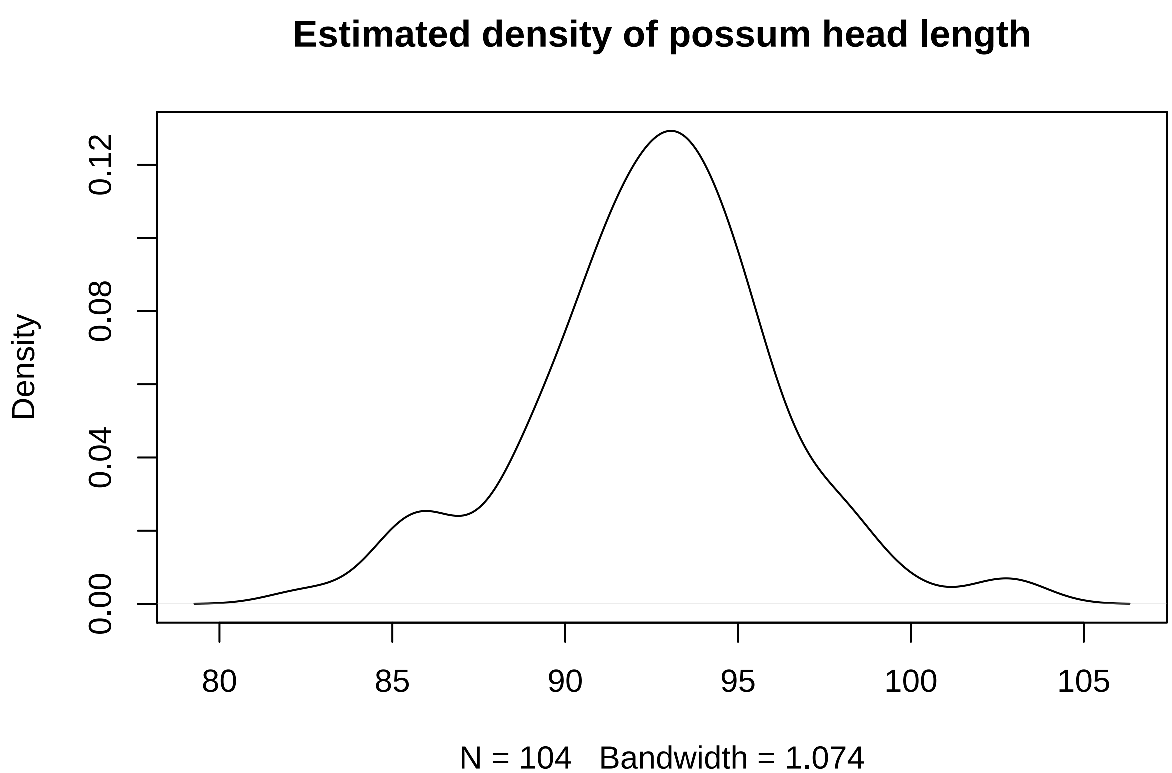
```r
# 3.3(b)
# Note: the code in the problem is incorrectly given as stem(qqnorm(possum$hdlength))
stem(possum$hdlngth)
```

```
##
##   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)
```



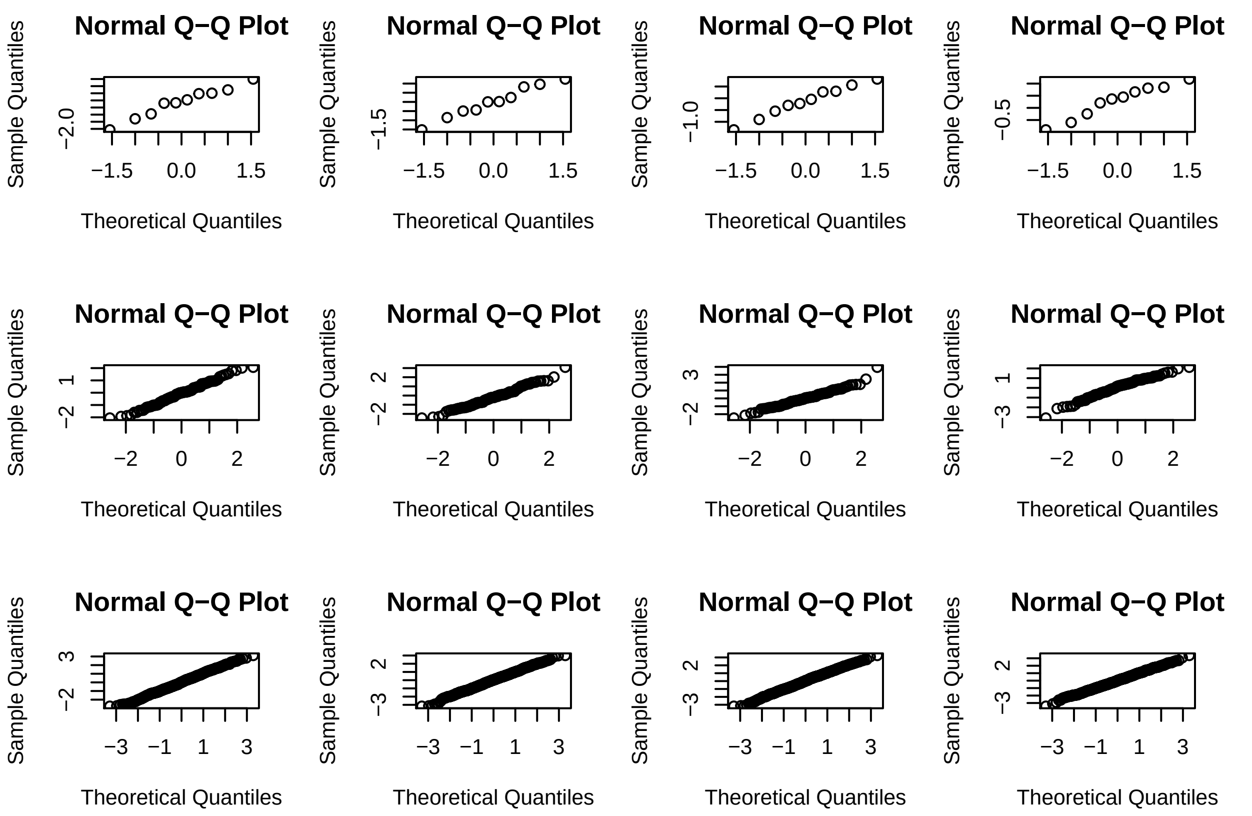

```r
# 3.3(d)
plot(density(possum$hdlngth), main = "Estimated density of possum head length")
```



These displays provide slightly different types of information. The stem and leaf plot is kind of a sideways histogram, with the bin widths fixed at 1, but it allows you to see all the individual data points. The histogram provides more flexibility in that bin widths are adjustable, but it is sensitive 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.